

# Computronic

## software für Heimcomputer

ÖS 55 Sfr 6,50 DM 6,50

### II Spiele

#### Mit Checksummer für:

- Commodore 64
- Commodore C 16
- Schneider CPC 464
- Atari
- VC-20

#### ● fehlerfreies Eingeben!

#### ● Top-Spiele für:

C64

C16

Schneider

ATARI

VIC-20

TI-99/4A

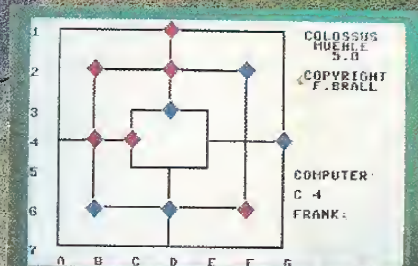
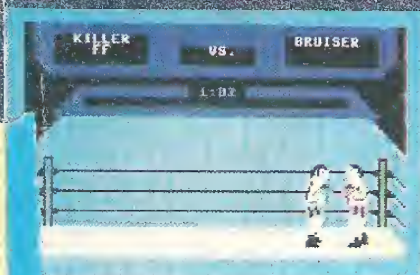
apple

Sinclair Spectrum

#### ● Tips & Tricks für C-16 Freunde

#### ● Traumland das erste Text-Adventure für den Schneider CPC-464

### BOXKAMPF



### COLOS.-MÜHLE

### BOULDER DASH





SA 2/86

# Compute mit

COMMODORE

## C=16

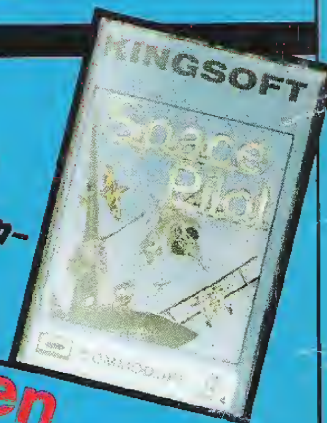
Heimcomputer

C=16 / plus 4

öS 55 sfr 6,50 DM 6,50

# SONDER- Ausgabe

50 mal können  
Sie diese Programm-  
Kassette gewinnen!



alle Programme mit Checksummen

Auch in dieser Ausgabe wieder  
interessante Beiträge zu den  
Themen:

- Werkstatt
- Bücher
- Tips & Tricks
- Hardware
- Software-Reviews
- u. v. m.

Sound-Machine – Key-Caller – Führung von Haushaltskonten – Textprogramm  
Pac-Mac – Jäger des verlorenen Schatzes – Oil of Texas – Skateboarding  
Mimi – Jäger der verlorenen Möhre – Soundhelfer – Auto-Starter – Ton-Clock



# Der neue Checksummer für alle C-16/116 - oder Plus 4-Besitzer!

Die Arbeitsweise des C-16/116 Checksummers entspricht weitgehend der VC-20- und C-64-Version und braucht deshalb nicht näher erklärt zu werden. Da auch der C-16/116 nicht den größten Speicher besitzt, wurde hier die Methode der VC-20-Version gewählt, und das Maschinenprogramm in den Kassettenpuffer gelegt. Dies hat den Vorteil, daß kein Basic-Speicher verlorengeht. Der Nachteil besteht darin, daß nach der Aktivierung des Checksummers keine Kassettenoperationen durchgeführt werden dürfen, da diese unweigerlich zum Systemabsturz führen. Um die Kassettenoperationen wieder zuzulassen, muß der Checksummer durch die beiden folgenden SYS-Aufrufe abgeschaltet werden:

**SYS 62158 : SYS 33047**

Aktiviert wird der Checksummer mit SYS 818. Die Funktion und Arbeitsweise entnehmen Sie bitte der C64-Checksummer-Version (siehe Ausgabe Computronic Nr. 3). Eine Ausnahme bei C-16/116 sind

die zusätzlichen Steuerzeichen 'FLASHON' (CTRL-), und 'FLASHOFF' (CTRL-) welche ebenfalls durch Klartext ersetzt werden (s. Tabelle Tastenbezeichnungen).

## Arbeitsweise und Aufbau unseres Checksummers:

Unser Checksummer besteht aus einem kleinen Maschinenprogramm welches als Basic-Loader abgedruckt ist.

Tippen Sie diesen Loader ein und speichern ihn auf Kassette oder Diskette, denn Sie können ihn zukünftig immer wieder benutzen.

Der Start erfolgt durch den Befehl „RUN“. Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC...“. Der Checksummer ist nun aktiv und man kann ein beliebiges Tronic-Listing eingeben. Vorher sollte der Checksummer mit dem Befehl „NEW“ gelöscht werden. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem Schirm eine Prüfzahl. Vergleichen Sie diese mit der

Zahl im Heft hinter der Zeile. Stimmt die Zahl überein, ist die Zeile richtig eingegeben.

Auf die Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben.

Interessant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen benutzt werden können, ohne die Checksumme zu verändern, Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben.

**Für alle, die nicht gerne abtippen, ist der Checksummer unter folgender Bestellnummer zu beziehen:**

**OV 10 K Kassette 10 DM  
OV 10 D Diskette 15 DM**

Fortsetzung Seite 48

```

1 REM ***** <4>
2 REM * * <99>
3 REM * C16-CHECKSUMMER * <140>
4 REM * * <101>
5 REM * COPYRIGHT BY FRANK BRALL * <182>
6 REM * * <103>
7 REM * (C) 18.04.86 * <181>
8 REM * * <105>
9 REM ***** <12>
10 PRINT "CLEAR DOWN SPACE2" + (SPACE2) <103>
CHECKSUMMER (SPACE) OC (SPACE) 1.0 (SPACE2) +
+ (DOWN) "
11 PRINT " (SPACE2) + (SPACE2) COPYRIGHT (SPA <110>
CE) FRANK (SPACE) BRALL (SPACE3) + (DOWN) "
12 PRINT " (SPACE8) FUER (SPACE) TRONIC-VERLA <196>
G"
60000 DIM H(75):FOR I=0 TO 9 <123>
60010 H(48+I)=I:H(65+I)=I+10:NEXT <145>
60020 FOR I= 818 TO 1010 :READ A$ <151>
60030 H=ASC (LEFT$(A$,1)) <234>
60040 L=ASC (RIGHT$(A$,1)) <253>
60050 D=H(H)*16+H(L):S=S+D:POKE I,D <13>
60060 A=A+1:IFA<20 THEN NEXT:A=-1 <97>
60070 READ V:Z=Z+1:IF V=S THEN 60085 <177>
60080 PRINT "DATAFEHLER (SPACE) IN (SPACE) ZE <39>
ILE (SPACE): ";60200+Z:END
60085 IFA<0 THEN 60100 <116>
60090 S=0:A=0:NEXT <246>
60100 PRINT " (DOWN2) " <172>
60110 PRINT "CHECKSUMMER (SPACE) EIN (SPACE) <140>
= (SPACE) SYS (SPACE) 818 (DOWN) "
60120 PRINT "CHECKSUMMER (SPACE) AUS (SPACE) <122>
= (SPACE) SYS (SPACE) 62158 + SYS (SPACE) 33047 (
DOWN) "
```

```

60130 PRINT " (DOWN) SAVE/LOAD (SPACE) IST (SP <61>
ACE) WAHREND (SPACE) DER (SPACE) CHECK- "
60140 PRINT "SUMMER (SPACE) AKTIV (SPACE) IST <221>
, (SPACE) NICHT (SPACE) MOEGLICH (SPACE) ! "
60150 PRINT " (DOWN) BEACHTEN (SPACE) SIE (SPA <189>
CE) DIE (SPACE) HINWEISE (SPACE) IN (SPACE) DEN
"
60170 PRINT "HEFTEN (SPACE) COMPUTE (SPACE) M <234>
IT (SPACE) UND (SPACE) COMPUTRONIC "
60180 PRINT " (DOWN) VIEL (SPACE) SPASS (SPACE <116>
3) ! (SPACE) (AUTOR: F. BRALL/6443SONTRA)
60190 POKE 814,147:POKE 815,251:POKE 816, <109>
147:POKE 817,251
60195 SYS 818:END <177>
60201 DATA A9,3D,8D,02,03,A9,03,8D,03,03 <56>
,60,A2,FF,86,3A,20,5A,8B,86,3B, 1851
60202 DATA 84,3C,20,73,04,AA,F0,EF,90,09 <28>
,20,53,89,20,79,04,4C,D9,8B,20, 2018
60203 DATA 3E,8E,20,53,89,84,0B,A9,00,8D <40>
,E4,03,8D,E5,03,8D,E6,03,18,A5, 2076
60204 DATA 14,65,15,8D,E6,03,A0,FF,C8,B1 <18>
,3B,F0,2C,C9,22,D0,0A,AD,E4,03, 2508
60205 DATA 49,01,8D,E4,03,B1,3B,AE,E4,03 <87>
,D0,04,C9,20,F0,E4,EE,E5,03,AE, 2644
60206 DATA E5,03,18,B1,3B,6D,E6,03,8D,E6 <107>
,03,CA,D0,F4,4C,76,03,38,20,39, 2204
60207 DATA D8,8E,E2,03,8C,E3,03,A2,00,A0 <115>
,00,18,20,39,D8,A9,5B,20,D2,FF, 2365
60208 DATA A9,00,AE,E6,03,20,5F,A4,A9,5D <169>
,20,D2,FF,20,08,FB,20,08,FB,AC, 2380
60209 DATA E3,03,AE,E2,03,18,20,39,D8,A2 <133>
,00,86,FF,4C,36,87,00,00,00,00, 1778
60210 DATA 00,00,00,00,00,00,00,00,00,00 <104>
,00,00,00, 0
```

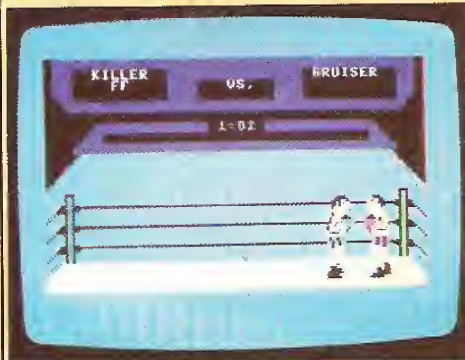


**C-64**

## FIGHT-NIGHT

„FIGHT-NIGHT“ ist wieder ein hervorragender Programmbeitrag in der großen Palette der Sportspiele des TRO-NIC-(Software)-Verlages.

Bei „FIGHT-NIGHT“ handelt es sich um die Darstellung eines spannenden Boxkampfes. Ihre Aufgabe besteht darin, die Kräfte Ihres Faustkämpfers sinn-



# TOP

voll einzuteilen, denn falls es Ihnen nicht gelingen sollte, Ihren Kontrahenten K.O. zu schlagen, entscheiden die Kampfrichter bzw. der Computer bei der Urteilsgebung nach Punkten. Und Sie als erfahrener Sportfreund wissen sicherlich, daß bei einem Punkturteil die letzte Runde, oft entscheidend ist. Also auf gehts - Ring Frei!

Seite 8

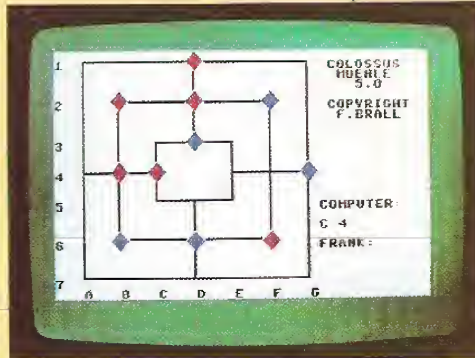
(S.G.)

**C-64**

## COLOSSUS MÜHLE

Dieses Programm ist sicherlich ein wertvoller und interessanter Beitrag im Bereich der Strategiespiele.

Bei „COLOSSUS MÜHLE“ sehen Sie sich einem Gegner gegenüber, der nur äußerst



schwer zu besiegen sein dürfte: denn Ihr Gegner heißt C-64!

Das Programm beruht auf dem Regeln des weit verbreiteten Mühlespiels. Das bedeutet mit anderen Worten, daß Ihr strategisches Denkvermögen gefordert wird und das gegen einen Gegner, der, falls Sie die Auseinandersetzung siegreich beenden wollen, Ihnen alles abverlangen wird. Und nun viel Spaß mit „COLOSSUS MÜHLE“ im Wettstreit gegen Ihren C-64-Computer.

Seite 19

(Frank Brall)

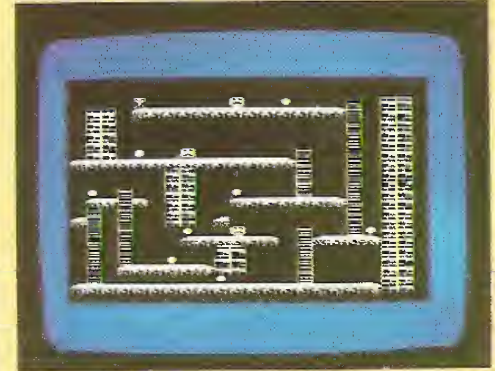
**VC-20**

## JOSEF IN DEN KATAKOMBEN

Dieses Programm läßt Sie eine Reise durch die Vergangenheit antreten. Am Ende Ihrer Reise angelangt, landen Sie in dem unterirdischen Labyrinthsystem einer Pyramide des „alten“ Ägyptens.

JOSEF als erfahrener Höhlenforscher läßt es sich nun natürlich nicht nehmen, die unterirdischen Gänge, die seit Jahrhunderten nicht mehr von Menschen betreten wurden, genauestens zu untersuchen. JOSEF scheint dabei eine „goldene Nase“ zu entwickeln, denn schon nach kurzer Zeit entdeckt er die sagenumwobenen Schätze des Pharaos Ramses II. Wird es JOSEF gelingen, diese Schätze an das Tageslicht zu befördern? Denn Sie können sich sicherlich denken, daß Ramses sich bestimmt ei-

niges zur Sicherung seiner Reichtümer hat einfallen lassen.



Seite 27

(S.G.)

**C-16**

## HUMOR

Daß der C-16-Computer mittlerweile zu einem Verkaufsschlager geworden ist, dürfte sich schon herumgesprochen haben. Natürlich trägt auch die Computronic-Redaktion dieser Entwicklung Rechnung. Wir werden auch weiterhin in jeder Computronic-Ausgabe mindestens zwei C-16-Programme veröffentlichen.

Hier stellen wir Ihnen nun das Programm „HUMOR“ vor. Der Titel dürfte anfangs vielleicht etwas irreführend sein, denn in diesem Spiel heißt es, den Verlust Ihres Hauses mit Humor zu tragen. Diesen Verlust können Sie nur vermeiden,



★★★ Was gibt's Neues in der Computronic ★★★

Basic Kurs (Teil 7) ... (Seite 7) Monsterjagd - Ein weiteres C-64-Programm (Seite 14) ... Impressum (Seite 35) ... Traumland - Das erste Schneider-Textadventure in der Computronic (S. 36) ... Jump-Man - Für Ihren Apple-Computer (Seite 70) ... Scootie - Ein toller Pro-

★★★ Was gibt's Neues in der Computronic ★★★



# Kurz belichtet

wenn es Ihnen gelingt die 9 versteckten Bomben rechtzeitig zu entschärfen und dabei nicht in die heimtückischen Fallen hineintappen.

Und nun viel Spaß und tragen Sie es mit Humor, falls Ihr Haus trotz aller Bemühungen doch explodieren sollte. Sie haben ja die Chance wieder von vorn zu beginnen.

Seite 33

(S.G.)

**CPC-464**

**TRITON**



Dieses Schneider-Programm stammt von Ottfried Schmidt einem unserer erfolgreichen Softwareautoren aus der verlagseigenen Programmierabteilung.

„TRITON“ ist ein drei-dimensionales Computerspiel, das vor allem durch seine fantastische Grafik, seine tolle Animation und die verwendeten 20 (!) Sprites besticht. Der Spieldauftrag besitzt folgenden Inhalt: Schlagen Sie, die von einem feindlichen Planeten initiierte Rebellion auf dem Planeten „Triton“ nieder und vernichten Sie die gegnerische Invasionsarmee.

(Ottfried Schmidt)

**Atari**

**BOULDER DASH**

„BOULDER DASH“ ist ein bekanntes Computerspiel für den C-64. Dieses Topprogramm können Sie jetzt dank Oliver Cyranka, der ebenso wie Zuheir Urwani, zu unseren bekanntesten Stammssoftwareautoren zählt, auch auf

**TOP**



Ihrem Atari-Computer (800XL/130XE) genießen.

Ihr Atari-Computer überträgt Ihnen dieses Mal den Posten eines „Steinbeißers“, der die mühevollen Aufgabe hat, die im Labyrinth verteilten Diamanten einzuheimsen. Mühevoll deshalb, weil Sie während Ihrer Diamantenjagd mit zahlreichen Gefahren wie Steinen, quadratischen Monstern, feindlichen Schmetterlingen und „schwabbelnden Massen“ konfrontiert werden.

All Ihre Bemühungen lohnen sich garantiert, denn bei „Boulder Dash“ handelt es sich um ein echtes Topprogramm.

Seite 49

(S.G.)

**ZX-Spectrum**

**ZONE-M**

„ZONE-M“ wird mit Sicherheit Ihre vollste Konzentration und schnellste Reaktionen erfordern. Die Spielaufgabe ist schnell erläutert: Sie müssen 13(!) Spielstufen möglichst schadlos überstehen, indem Sie sich erfolgreich den immer zahlreicher werdenden Gegner erwehren. Sollte es Ihnen nicht gelingen Ihre Widersacher aus dem Verkehr zu ziehen, so kommt es zu einer ernsthaften Gefährdung Ihres „Spielerdaseins“.

„ZONE M“ stammt aus der Feder

von Zuheir Urwani aus Recklinghausen, einem unserer profiliertesten Stammsautoren und kann mit Fug und Recht als Toppprogramm bezeichnet werden.

**TOP**



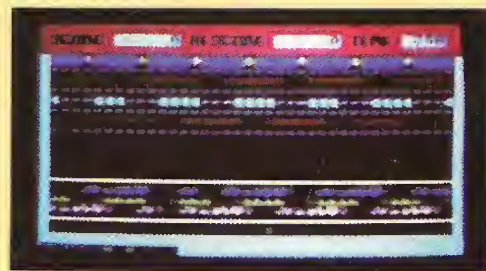
Seite 65

(S.G.)

**C-16**

**FROGGER**

Noch ein weiteres Programm für den Commodore-Verkaufsschlager.“ Werner Bembon, der Softwareautor von Frogger läßt



Sie in diesem Spiel Frösche sicher über den Bildschirm steuern. Ein Vorhaben, das sich im Nachhinein als äußerst schwierig erweist.

Seite 29

(S.G.)

★★★ Was gibt's Neues in der Computronic ★★★

grammbeitrag für alle TI-99-Besitzer (S. 74) ... Tips und Tricks für den C-16 und VC-20 (S. 79) ... Büchermarkt (Seite 82) ... Tips und Tricks für den Atari (Seite 84) ... Software-champion '86 (Seite 84) ... Rätsel (Seite 85) ... Großer Softwarekatalog (Seite 86/87) ...

★★★ Was gibt's Neues in der Computronic ★★★





# ● BASIC-KURS Teil 7 ●

Nachdem wir in der letzten Computronic-Ausgabe mit dem Kapitel Stringverarbeitung begonnen haben, sind wir jetzt in der Lage erste nützliche Programme zu entwickeln. In der 7. Folge unseres Basic-Kurses stellen wir nun weitere wichtige Stringfunktionen vor.

## Konvertierungs- und Zeichenkettenfunktionen

Wie wir schon in den ersten beiden Teilen unseres Kurses gelernt haben, entspricht jedes dargestellte Zeichen, sowie auch alle Buchstaben und Zahlen, einem speziellen Code: Dem ASCII-Code. Da es bei speziellen Anwendungen von Vorteil ist, statt mit dem Zeichen, mit dem Code zu arbeiten, kennt der Basic-Interpreter zwei Umwandlungs-Befehle. Dadurch ist es beispielsweise relativ einfach, Namen nach dem Anfangsbuchstaben zu sortieren, da den Buchstaben im ASCII-Code aufsteigende Werte zugeordnet sind.

### ASC (...)

Die ASC-Funktion ermöglicht es, den internen Zahlenwert eines Buchstabens oder eines anderen Zeichen's zu ermitteln. Tippen wir einmal das folgende Beispiel in unseren Rechner:

```
10 INPUT "EINEN BUCHSTABEN
   EINGEBEN";B$
20 W=ASC(B$)
30 PRINT "DER BUCHSTABEN "B$
   BESITZT DEN ASCII-WERT "B
40 GOTO 10
```

Wir starten unser Beispiel:

```
RUN <RETURN>
A <RETURN>
65
B <RETURN>
66
7 <RETURN>
55
```

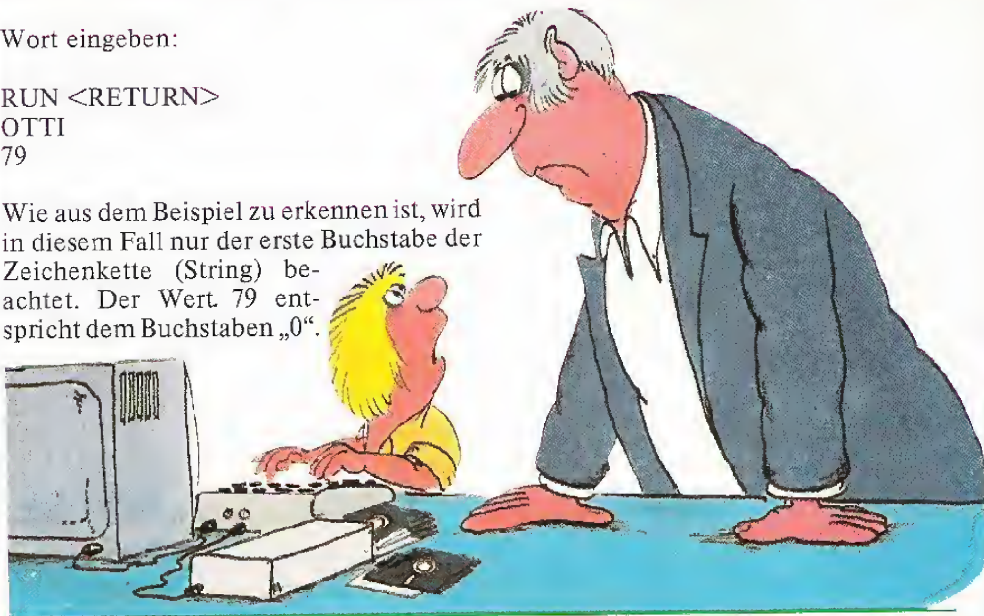
Wenn wir in der ASCII-Tabelle (Teil 1) nachschauen, so werden wir erkennen, was die ausgegebenen Zahlen zu bedeuten haben. Der ASCII-Wert vom Buchstaben „A“ beträgt somit 65, der von „B“ beträgt 66 usw.

Interessant ist es auch, wenn wir einmal statt einem Buchstaben ein komplettes

Wort eingeben:

```
RUN <RETURN>
OTTI
79
```

Wie aus dem Beispiel zu erkennen ist, wird in diesem Fall nur der erste Buchstabe der Zeichenkette (String) beachtet. Der Wert 79 entspricht dem Buchstaben „O“.



*Mensch Vati, Du immer mit den „blöden“ Hausaufgaben, der Basic-Kurs in Computronic ist doch viel wichtiger!*

### CHR\$ (...)

Die entgegengesetzte Funktion zu ASC ist die CHR\$-Funktion. Diese ermittelt aus einer Zahl das alphanumerische Zeichen. So können wir aus dem Wert 79 wieder eine „O“ werden lassen:

```
PRINT CHR$(79) <RETURN>
O
```

Leider besitzen im ASCII-Code nur die Zahlenwerte von 32 bis 127 allgemeine Gültigkeit zur Darstellung von alphanumerischen Zeichen. Die Zahlenwerte von 0 bis 31 stellen in der Regel Steuerzeichen dar. Da diese jedoch nicht genormt sind, verwenden die einzelnen Computer unter-

schiedliche Vereinbahrungen. Auch die Zahlenwerte von 128 bis 255 sind nicht genormt, werden jedoch häufig für Grafik- oder reverse Zeichen benutzt. Mit dem folgenden Programm können wir uns eine komplette ASCII-Code Tabelle auf dem Schirm ausgeben lassen. Die Werte unter 32 und über 127 werden aus den erwähnten Gründen nicht beachtet. Trotz dieser Einschränkung ist das Beispiel nicht auf allen Rechnern lauffähig, da einige Computer von der ASCII-Norm abweichen und ebenfalls Steuerzeichen in diesen Bereich unterbringen. Ein Beispiel sind die Commodore-Rechner, die nach der Ausgabe eines Anführungszeichens oft nicht mehr fehlerfrei arbeiten.

```
10 REM BEISPIEL 7.1
100 PRINT "ASCII-TABELLE COMPUTRONIC-KURS"
110 PRINT "-----"
120 PRINT
130 FOR Z= 0 TO 15
140 FOR S= 2 TO 7
150 IF S*16+Z>95 THEN 180
160 PRINT TAB(S*6-11);S*16+Z;TAB(S*6-7);CHR$(S*16+Z);
170 GOTO 190
180 PRINT TAB(S*7-17);S*16+Z;TAB(S*7-12);CHR$(S*16+Z)
190 NEXT S
200 PRINT
210 NEXT Z
READY.
```





## Das erste sinnvolle Programm:

### Umrechnung römischer Zahlen

Wir haben inzwischen die wichtigsten Eigenschaften und Möglichkeiten eines Basic-Interpreters kennengelernt. Wir wollen nun unsere Kenntnisse dazu verwenden, ein kleines Umwandlungsprogramm zu schreiben.

Das römische Zahlensystem ist nicht wie unser Dezimalsystem ein Stellenwertsystem mit einer bestimmten Wertigkeit je Stelle, sondern die einzelnen Ziffern haben einen Wert:

I = 1  
V = 5  
X = 10  
L = 50  
C = 100  
D = 500  
M = 1000

Bei der Umrechnung in das Dezimalsystem müssen wir deshalb werten, ob bei der römischen Zahl benachbarte Ziffern größer oder aber kleiner sind. Denn nur so kann man entscheiden, ob der Wert der Ziffer addiert oder subtrahiert wird.

Mit der ersten FOR...NEXT Schleife wird die römische Zahl „zerlegt“, und die in ihr vorkommenden Wertigkeiten werden dem Feld A zugeordnet. In der zweiten Schleife wird dann die Dezimalzahl ermittelt, indem man aufgrund der Anfrage in Zeile 280 entscheidet, ob addiert oder aber subtrahiert wird.

Um Ihr Verständnis zu prüfen, sollten Sie sich einmal die Funktion an der römischen Zahl MCMLXXXIV klarmachen.

### Rechnen mit Buchstaben und Zeichen

Neben den bisher aufgezeigten Stringfunktionen gibt es noch einige weitere interessante Möglichkeiten. Da es öfters vorkommt, daß in einer Stringvariable eine Zahl enthalten ist, existiert eine Funktion, die dieser Zahl einem String entnimmt und einer Zahlen-Variable zuweist. Dies ist die VAL-Funktion.

```
PRINT VAL("100") <RETURN>
100
```

Auf den ersten Blick scheint es, als habe die VAL-Funktion keine Bedeutung, denn die

#### SEARCHING FOR BASIC72

```
100 REM ROEMISCHE ZAHLEN
110 REM BEISPIEL 7.2
120 REM
130 INPUT "ROEMISCHE ZAHL ";R$
140 L=LEN(R$)
150 FOR I=1 TO L
160 Z$=MID$(R$,I,1)
170 IF Z$="I" THEN A=1
180 IF Z$="V" THEN A=5
190 IF Z$="X" THEN A=10
200 IF Z$="L" THEN A=50
210 IF Z$="C" THEN A=100
220 IF Z$="D" THEN A=500
230 IF Z$="M" THEN A=1000
240 A(I-1)=A
250 NEXT I
260 REM DEZIMALZAHL BERECHNEN
270 FOR I=L-1 TO 0 STEP-1
280 IF A(I)<A(I+1) THEN D=D-A(I):GOTO 290
285 D=D+A(I)
290 NEXT I
300 REM ERGEBNISAUSGABE
310 PRINT R$;" =";D
320 END
READY.
```

Zahl 100 wäre auch ohne diese Funktion ausgegeben worden. Das nächste Beispiel zeigt schon etwas deutlicher die Funktion von VAL:

```
PRINT VAL("100DM") <RETURN>
100
```

Wir wollen auch hier ein kleines Beispielprogramm schreiben:

```
10 REM Beispiel 7.3
20 INPUT "STRING ";S$
30 Z=VAL(S$)
40 PRINT "DIE ZAHL LAUTET "Z
50 GOTO 20
```

Wir starten ...

```
RUN <RETURN>
STRING ? ICH HABE 100 DM
DIE ZAHL LAUTET 0
STRING ? 100 DM HABE ICH
DIE ZAHL LAUTET 100
STRING ? 123.45 DM SCHULDEN
123.45
```

Wie zu erkennen ist, löst VAL nur die Zahlen heraus, welche direkt am Anfang des String liegen. Weitere eventuell auftretende Zahlen werden im String nicht berücksichtigt. Das ganze können wir auch umdrehen. In Basic existiert ebenfalls eine Funktion zum Umwandeln einer Zahl oder Zahlenvariable in einen String. Diese Funktion heißt STR\$ und wird wie folgt angewendet:

```
PRINT STR$(1986)
1986
```

Auch bei diesem Beispiel ist auf den ersten Blick keine Funktion zu erkennen, da eine Zahl sich im Aussehen nicht von einem String unterscheidet. Die nachfolgende Programmzeile zeigt die Funktion etwas deutlicher:

```
PRINT STR$(1986)+"TEST"
1986TEST
```

Es versteht sich von selbst, daß eine mit STR\$ umgewandelte Zahl danach mit al-



len anderen Stringfunktionen wie LEFT\$, RIGHT usw. bearbeitet werden kann. Sie werden sich jetzt sicher fragen, wozu die Umwandlungen eigentlich gut sind. Ein Anwendungsfall wäre beispielsweise die Entwicklung einer benutzerfreundlichen und abbruchsicheren Eingaberoutine. Wir werden uns damit in einer der nächsten Folgen unseres

Basic-Kurses noch genauer beschäftigen.

**Literaturquellen:**

Grundkurs in Basic, Ulrich Strobel, Sybex Verlag  
Basic Brevier, Siegmund Wittig, Heinz Heise Verlag

Commodore Handbücher  
Spectrum Handbuch  
Commodore 116/16/plus4,  
Ekkehard Kaier,  
Vieweg Verlag

Basic Kurs Teil 8 folgt in der nächsten Computronic!

Der



COMMODORE-

SONDERTEIL

FIGHT NIGHT

Ein attraktives  
Sportprogramm  
für Ihren Commodore 64

Fight Night ist ein Boxspiel für 2 Personen. Der spannende Kampf geht über 10 Runden. In den Rundenpausen werden die Punkte angezeigt. Und zwar die Punkte der letzten Runde, die Gesamtpunktzahl, und die Kraft, die der Boxer noch zur Verfügung hat. Ist diese kleiner als Null, geht der Boxer K.O.

Zuletzt wird noch die Ausdauer angegeben, die sich mit jedem Schlag verringert. Deshalb ist es nicht sinnvoll, wild auf seinen Gegner einzuschlagen. Denn läßt die Ausdauer nach, muß sich der Boxer zwischen den Schlägen länger erholen. Sollten nach 10 Runden beide Boxer noch kampffähig sein, wird der Sieger nach Punkten bestimmt.

**Steuerung:**

Die Figuren werden mit den beiden Joysticks gesteuert. Dabei sind folgende Bewegungen möglich:  
links-rechts gehen (Joystick in die entsprechende Richtung)  
oben/unten decken (Joystick oben und unten)  
oben/unten schlagen (Joystick oben/unten + Feuerknopf)  
Als Treffer zählt nur der Schlag, der die ungedeckte Seite des Gegners trifft.

**Action**

**FIGHT NIGHT** ist ein echtes Topprogramm! Hier wird nicht wildes „Drauflosdreschen“ gefordert, sondern taktisch geschicktes Kampfverhalten und kluges Einteilen der Kräfte verlangt.

**Spannung/Dramatik/aufregendes Kampfgeschehen**

**Ring frei! Sie sind der Champion und müssen Ihre Ausdauer und Geschicklichkeit einsetzen, um Ihren Titel zu verteidigen.**





```

RVSOFF}L" <195>
1025 PRINT "{SPACE2}{RVSON SPACE34 RVSOFF  
E}L" <149>
1030 PRINT "{SPACE}{RVSON SPACE36 RVSOFF  
}L" <157>
1040 PRINT "{RVSON SPACE38 RVSOFF}L"; <100>
1045 PRINT "{RVSON SPACE2 RVSOFF ORANGE}"  
{RVSON CYAN SPACE34 ORANGE RVSOFF}{CYAN  
RVSON SPACE2 RVSOFF}"; <95>  
1050 FORA=0T02 <87>
1055 PRINT "{CYAN RVSON SPACE RVSOFF ORAN  
GE}|_|||||_|-  
|{CYAN RVSON SPACE RVSOFF}"; <183>
1060 PRINT "{ORANGE}*{CYAN RVSON SPACE RV  
SOFF ORANGE}-{RVSON CYAN SPACE34 RVSOFF  
ORANGE}-{RVSON CYAN SPACE RVSOFF ORANGE}  
!"; <109>
1065 NEXT <175>
1070 PRINT "{RVSON CYAN SPACE RVSOFF HGRE  
Y}-{RVSON SPACE36 RVSOFF HGREY}-{CYAN RV  
SON SPACE RVSOFF}"; <108>
1075 PRINT "{HGREY}-{RVSON SPACE38 RVSOFF  
HGREY}-"; <136>
1080 PRINT "{YELLOW RVSON SPACE40 RVSOFF}  
"; <86>
1090 FORA=960T0999:POKE34816+A,160:POKE5  
5296+A,3:NEXT <214>
1095 FORA=0T039:POKE34816+A,160:POKE5529  
6+A,3:NEXT:POKE646,1 <159>
1100 POKEV+17,155 <76>
1135 REM <2>
1140 REM KAMPF VORBEREITEN <200>
1145 REM <12>
1150 POKEV+21,0:POKEV+23,255:SYS28762 <77>
1152 POKEV+4,54:POKEV+8,48:POKEV+5,158:F  
OKEV+9,200:POKEV+1,158:POKEV+3,158 <221>
1155 POKEV+6,32:POKEV+10,38:POKEV+7,158:  
POKEV+11,200:POKEV+16,40 <235>
1160 POKE35834,48:POKE35836,53:POKE35835  
,57:POKE35837,62 <130>
1165 POKEV+41,5:POKEV+43,5:POKEV+42,4:P  
OKEV+44,4:POKEV+39,5:POKEV+40,4 <87>
1170 SI=54272:POKESI+7,0:POKESI+8,50:POK  
ESI+11,0:POKESI+12,9:POKESI+13,0 <152>
1175 POKESI+24,15:POKESI+14,0:POKESI+15,  
35:POKESI+18,0:POKESI+19,10:POKESI+20,0 <196>
1180 RA$(0)="{SPACE}BUM{SPACE}BUM{SPACE2  
3":RA$(1)="{SPACE}THUNDER{SPACE2}":RA$(2  
)="{SPACE2}KILLER{SPACE2}" <200>
1181 RA$(3)="{SPACE}BRUISER{SPACE2}" <234>
1183 IFDEMO=1THENRETURN <253>
1200 SI=40:S2=40:RU=1:POKE49179,50:POKE4  
9180,50:G1=0:G2=0:POKEV+21,60 <253>
1300 FORA=1T02:POKE211,3+(A-1)*24:POKE21  
4,3:SYSS8732:N$(A)="" :X=0:PRINT "{WHITE}"  
; <76>
1303 GETA$:PRINT "{RVSON SPACE RVSOFF LEF  
T}"; <88>
1305 IFA$=CHR$(20) THENIFX>0THENPRINT "{LE  
FT SPACE2 LEFT2}":N$(A)=LEFT$(N$(A),LEN  
(N$(A))-1) <78>
1306 IFA$=CHR$(20) THENX=X-1 <217>
1310 IFA$>="A"AND A$<="Z"ORA$="{SPACE}"TH  
ENPRINTA$:N$(A)=N$(A)+A$:X=X+1 <182>
1315 IFA$=CHR$(13) THENPRINT "{SPACE}";:GO  
TO1325 <115>
1320 IFX<10THENGOTO1303 <66>
1325 POKE211,(A^1)*24+3:SYSS8732:PRINT "{  
SPACE10}"; <129>
1327 POKE211,5-INT(LEN(N$(A)))/2+.5)+3+(A  
-1)*24:SYSS8732:PRINTN$(A) <79>
1330 N1$(A)=RA$(RND(O)*4):POKE211,A*24-2  
1:POKE214,2:SYSS8732:PRINTN1$(A):NEXT <83>

```





# Commodore 64



```

1500 POKE49156,27:POKE49158,24:POKE49157
,144:POKE49159,147 <81>
1505 POKE49178,32:POKE49177,0:POKE49175,
0:POKE49176,0:POKE49186,0:POKE49187,0 <202>
1510 POKE49171,2:POKE49172,0:POKE49173,1
:POKE49152,0:POKE49154,0 <22>
1550 X=16:Y=3:Z$="WHITE SPACE3"VS.(SPAC
E2)":GOSUB10000:Y=4:Z$="(SPACE8)":GOSUB1
0000 <240>
2000 POKESI+18,0:POKESI+18,21:POKE49183,
S1:POKE49184,S2 <60>
2002 SYS28844:POKE56322,255 <36>
2005 IFPEEK(49186)=255ORPEEK(49187)=255T
HEN5000 <236>
2985 REM <67>
2990 REM RUNDENBEWERTUNG <213>
2995 REM <77>
3000 POKESI+18,0:POKESI+18,21 <133>
3001 T1=PEEK(49175):T2=PEEK(49176):POKE6
46,7:IFT1=T2THENP1=10:P2=10:GOTO3012 <214>
3002 POKE35834,48:POKE35835,57:POKEV+21,
60 <170>
3005 IFABS(T1-T2)<4THENP1=9.5+SGN(T1-T2)
/2:P2=17-P1 <8>
3010 IFABS(T1-T2)>3THENP1=9+SGN(T1-T2):P
2=18-P1 <151>
3012 POKEV+21,60:POKE35834,48:POKE35835,
57 <37>
3015 X=16:Y=3:C=1:Z$="(SPACE)ROUND(SPACE
2)":GOSUB10000:Z$=STR$(RU):GOSUB11000:RU
=RU+1 <43>
3017 X=18:Y=4:C=7:GOSUB10000:FORA=0TO300
0:NEXT <239>
3020 P1$=STR$(P1):Z$=P1$:GOSUB11000:P1$=
Z$:Z$=STR$(P2):GOSUB11000:P2$=Z$ <105>
3025 X=6:Y=4:C=7:Z$=P1$:GOSUB10000:X=30:
Z$=P2$:GOSUB10000 <199>
3030 X=16:Y=3:C=1:Z$="JUDGES"(SPACE)":GO
SUB10000:Y=4:Z$="(SPACE)SCORE(SPACE2)":G
OSUB10000 <139>
3035 FORA=0TO2000:NEXT <121>
3045 G1=G1+P1:G2=G2+P2:G1$=STR$(G1):G2$=
STR$(G2) <127>
3047 Z$=G1$:GOSUB11000:G1$=Z$:Z$=G2$:GOS
UB11000:G2$=Z$ <244>
3050 X=6:Y=4:C=7:Z$=G1$:GOSUB10000:X=30:
Z$=G2$:GOSUB10000 <21>
3055 X=16:Y=3:C=1:Z$="(SPACE)TOTAL(SPACE
2)":GOSUB10000:Y=4:Z$="(SPACE)SCORE(SPA
CE2)":GOSUB10000 <233>
3060 FORA=0TO2000:NEXT:S1=INT(S1-T2/2+A1
/10):S2=INT(S2-T1/2+A2/10) <84>
3065 Z$=STR$(S1):GOSUB11000:S1$=Z$:Z$=ST
R$(S2):GOSUB11000:S2$=Z$ <251>
3070 X=6:Y=4:C=7:Z$=S1$:GOSUB10000:X=30:
Z$=S2$:GOSUB10000 <141>
3075 X=16:Y=3:C=1:Z$="STRENGTH":GOSUB100
00:Y=4:Z$="(SPACE2)LEFT(SPACE2)":GOSUB10
000 <169>
3080 FORA=0TO2000:NEXT:A1=INT(PEEK(49179
)+5):A2=INT(PEEK(49180)+5) <179>
3081 IF A1>200THENA1=0 <52>
3082 IF A2>200THENA2=0 <66>
3083 IF A1>50THENA1=50 <113>
3084 IF A2>50THENA2=50 <126>
3085 POKE49179,A1:POKE49180,A2 <37>
3087 Z$=STR$(A1):GOSUB11000:A1$=Z$:Z$=ST
R$(A2):GOSUB11000:A2$=Z$ <154>
3090 X=6:Y=4:C=7:Z$=A1$:GOSUB10000:X=30:
Z$=A2$:GOSUB10000 <12>
3097 X=16:Y=3:C=1:Z$="STAMINA(SPACE)":GO
SUB10000:Y=4:Z$="(SPACE2)LEFT(SPACE2)":G
OSUB10000 <150>
3100 POKE211,6:POKE214,8:SYS58732:PRINT"
(SPACE28)" <219>

```

```

3105 FORA=0TO2000:NEXT:X=6:Y=4:Z$="(SPAC
E3)":GOSUB10000:X=30:GOSUB10000 <230>
3107 X=16:Y=3:Z$="WHITE SPACE3"VS.(SPAC
E2)":GOSUB10000:Y=4:Z$="(SPACE8)":GOSUB1
0000 <11>
3110 X1=SGN(27-PEEK(49156)):X2=SGN(144-P
EEK(49157)) <75>
3120 IFPEEK(49156)<>27THENPOKE49156,PEEK
(49156)+X1:POKE49158,PEEK(49158)+X1 <171>
3122 IFPEEK(49157)<>144THENPOKE49157,PEE
K(49157)+X2:POKE49159,PEEK(49159)+X2 <84>
3125 SYS29789:IFPEEK(49156)<>27ORPEEK(49
157)<>144THENGOTO3120 <68>
3200 IFRU<11THENGOTO1500 <149>
3985 REM <47>
3990 REM KAMPF BEENDET <206>
3995 REM <57>
4000 XX=3:E=1:IFB1<62THENXX=27:E=2 <81>
4005 IFB1=62THENIFB2>S1THENXX=27:E=2 <171>
4007 X1=5-INT(LEN(N$(E))/2+.5)+XX <234>
4010 X=16:Y=3:C=1:Z$="(SPACE)WINNER(SPA
CE3)":GOSUB10000:Y=4:Z$="(SPACE3)IS(SPA
CE3)":GOSUB10000 <88>
4015 FORA=0TO10:FORB=0TO1:FORC=0TO500:NE
XT <206>
4020 X=XX:Y=2:C=B+1:Z$=N1$(E):GOSUB10000
:X=X1:Y=3:Z$=N$(E):GOSUB10000 <155>
4025 NEXT:Z$="(SPACE10)":X=3:Y=2:GO
SUB10000:X=27:GOSUB10000 <137>
4030 Y=3:GOSUB10000:X=3:GOSUB10000:GOTO1
50 <249>
4985 REM <27>
4990 REM K.O. <98>
4995 REM <37>
5000 POKEV+16,PEEK(V+16)AND252:FORA=0TO1
50:NEXT:IFPEEK(49186)=255THEN5500 <18>
5005 SYS29839:SYS29854:POKEV+1,172:POKEV
+5,172:POKEV+9,214:POKE35835,57 <113>
5010 X=PEEK(V+8) <193>
5015 POKEV+8,X:POKEV+4,X:X=X-24 <139>
5017 POKEV+21,(PEEK(V+21)OR1)AND253 <111>
5020 POKEV,X:POKE35832,24:POKE35834,25:P
OKE35836,26:FORA=0TO200:NEXT <116>
5030 POKEV+9,208:POKEV+5,208:POKEV+1,208
:POKE35832,28:POKE35836,29 <28>
5035 X=X-24:IFX<0THENX=X+256:POKEV+16,PE
EK(V+16)AND251 <82>
5040 POKE35834,27:POKEV+4,X:E=2:XX=27:GO
TO6000 <92>
5500 SYS29874 <105>
5502 SYS29839:POKEV+3,172:POKEV+7,172:PO
KEV+11,214:POKE35834,48 <188>
5505 X=PEEK(V+10):IFPEEK(V+16)AND32THENP
OKEV+16,PEEK(V+16)OR10 <62>
5510 POKEV+10,X:POKEV+6,X:X=X+24:IFX>255
THENX=X-256:POKEV+16,PEEK(V+16)OR2 <48>
5520 POKEV+21,(PEEK(V+21)OR2)AND254 <141>
5525 POKEV+2,X:POKE35833,24:POKE35835,25
:POKE35837,26:FORA=0TO200:NEXT <164>
5530 POKEV+11,208:POKEV+7,208:POKEV+3,20
8:POKE35833,28:POKE35837,29 <182>
5535 X=X+24:IFX>255THENX=X-256:POKEV+16,
PEEK(V+16)OR8 <28>
5540 POKE35835,27:POKEV+6,X:E=1:XX=3 <59>
5545 IFDEM=1THENRETURN <24>
6000 POKE211,6:POKE214,8:SYS58732:PRINT"
(SPACE28)" <58>
6005 FORA=0TO2000:NEXT:GOTO4007 <27>
9985 REM <183>
9990 REM INTERPROGRAMME <238>
9995 REM <193>
10000 POKE211,X:POKE214,Y:SYS58732:POKE6
46,C:PRINTZ$:RETURN <173>
11000 IFVAL(Z$)<10THENZ$="(SPACE)0"+MID$
(Z$,2,1):RETURN <49>

```



```

11005 RETURN
19985 REM
19990 REM      INTRO
19995 REM
20000 DEMO=1:POKEV+23,255:GOSUB1152:POKE
V+21,60:POKEV+16,0
20002 POKE49156,7:POKE49158,4:POKE49157,
162:POKE49159,165:SYS29789
20005 POKESI+24,15:POKESI+6,0:POKESI+1,5
20010 PRINT"CLEAR DOWN2 SPACE6 BLUE RV
SON:"{RVSOFF}"{RVSON}"{RVSOFF SPACE RVSO
N}"{RVSOFF SPACE}"{RVSON}"{RVSOFF}"{RV
SON}"{RVSOFF}"
20015 PRINT"{SPACE6 RVSON}"{RVSOFF}"{RV
SON}"{RVSOFF}"{RVSON}"{RVSOFF SPACE
RVSON}"{RVSOFF SPACE2 PURPLE RVSON}"{RV
SOFF}"{RVSON}"{RVSOFF SPACE}"{RVSON}"{RV
SOFF}"{RVSON}"{RVSOFF SPACE}"{RVSON}"{R
VSOFF}"
20020 PRINT"{BLUE SPACE6}"{SPACE2}"{RV
SON}"{RVSOFF SPACE}"{SPACE}"{SPACE2 PUR
PLE RVSON}"{RVSOFF}"{RVSON}"{RVSOFF}"{R
VSON}"{RVSOFF}"{SPACE RVSON}"{RVSOFF}"
20025 PRINT"{SPACE20 PURPLE}"{RVSON}"{R
VSOFF SPACE}"{RVSON}"{RVSOFF SPACE}"{RV
SON}"{RVSOFF SPACE RVSON}"{RVSOFF}"
20030 PRINT"{DOWN BLACK SPACE18}BY"
20035 PRINT"{DOWN SPACE12}THOMAS{SPACE}G
DESMANN"
20100 FORA=0TO67:POKE49156,PEEK(49156)+1
:POKE49158,PEEK(49158)+1
20105 POKE49157,PEEK(49157)-1:POKE49159,
PEEK(49159)-1:SYS29789:NEXT
20107 POKE35835,58:FORA=0TO100:NEXT
20110 FORB=0TO1:POKE35835,58:POKEV,174:P
OKEV+21,61:POKE35832,51:POKE35834,50
20112 POKESI+4,0:POKESI+5,3:POKESI+4,129
20115 FORA=0TO300:NEXT:POKEV+21,60:POKE3
5834,48:FORA=0TO400:NEXT:NEXT
20120 POKEV+21,61:POKE35832,56:POKE35834
,55:POKESI+4,0:POKESI+5,8:POKESI+4,129
20125 PRINT"CLEAR":GOSUB5500:FORB=0TO1
000:NEXT:POKEV+21,0:POKEV+33,0:DEMO=0
20130 POKEV+32,3:RETURN
39985 REM
39990 REM      M-CODE DATAS
39995 REM
40000 DATA169,149,141,0,221,169,33,141,2
4,208,169,136,141,136,2,169,0,133,139
40001 DATA169,136,133,140,160,0,169,32,1
45,139,136,208,251,230,140,165,140
40002 DATA201,141,208,241,169,127,141,14
,220,169,51,133,1,169,0,133,139,133
40003 DATA141,169,208,133,140,169,128,13
3,142,162,8,160,0,177,139,145,141,136
40004 DATA208,249,230,140,230,142,202,20
8,242,169,55,133,1,169,129,141,14,220
40005 DATA120,169,105,141,20,3,169,112,1
41,21,3,89,96,0,0,162,2,254,255,191
40006 DATA189,255,191,201,24,208,33,169,
0,157,255,191,254,1,192,188,1,192,192
40007 DATA4,208,5,160,0,156,1,192,185,24
0,111,224,2,208,3,185,244,111,157,251
40008 DATA139,202,234,208,209,76,183,113
,0,120,169,49,141,20,3,169,234,141
40009 DATA21,3,88,96,0,169,224,141,2,220
,162,2,160,4,189,255,219,133,2,165
40010 DATA2,41,18,208,3,76,16,114,165,2,
41,17,208,3,76,129,114,32,240,113,165
40011 DATA2,41,1,208,3,32,221,113,165,2,
41,8,208,3,76,144,113,165,2,41,4,208
40012 DATA3,76,22,113,165,2,157,12,192,1
89,10,192,240,3,222,10,192,189,16,192
40013 DATA240,3,222,16,192,202,136,136,2

```

<181>  
<238>  
<51>  
<248>

<121>

<70>

<208>

<194>

<166>

<75>

<60>

<212>

<24>

<4>

<222>

<157>

<167>

<108>

<60>

<157>

<156>

<50>

<92>

<244>

<102>

<122>

<0>

<228>

<13>

<144>

<226>

<99>

<116>

<211>

<120>

<170>

<8>

<41>

# FIGHT NIGHT DAS LISTING

```

08,177,162,48,160,0,136,208,253,202
40014 DATA208,250,32,168,115,76,177,112,
0,0,189,3,192,201,32,144,27,224,2,208
40015 DATA11,173,5,192,56,237,4,192,201,
16,144,12,222,3,192,222,14,192,222
40016 DATA5,192,32,61,113,76,234,112,0,0
,173,16,208,61,225,111,141,16,208,189
40017 DATA3,192,10,153,2,208,144,9,173,1
6,208,29,233,111,141,16,208,173,16
40018 DATA208,61,223,111,141,16,208,189,
14,192,10,153,254,207,144,9,173,16
40019 DATA208,29,231,111,141,16,208,173,
16,208,61,227,111,141,16,208,189,5
40020 DATA192,10,153,6,208,144,9,173,16,
208,29,235,111,141,16,208,96,0,189
40021 DATA3,192,201,141,176,27,224,1,208
,11,173,5,192,56,237,4,192,201,16,144
40022 DATA12,254,3,192,254,14,192,254,5,
192,32,61,113,76,234,112,0,0,162,2
40023 DATA160,4,254,7,192,189,7,192,201,
16,208,14,169,0,157,7,192,169,61,56
40024 DATA249,3,208,153,3,208,202,136,13
6,208,227,76,49,234,0,0,189,10,192
40025 DATA208,11,169,49,224,2,208,2,169,
58,157,249,139,96,0,0,189,10,192,240
40026 DATA4,234,234,234,96,169,48,224,2,
208,2,169,57,157,249,139,173,21,208
40027 DATA61,223,111,141,21,208,96,0,0,1
89,12,192,41,18,240,101,189,16,192
40028 DATA208,96,234,224,2,208,22,189,3,
192,56,233,12,157,14,192,169,30,141
40029 DATA251,139,169,31,141,249,139,76,
74,114,169,55,141,250,139,169,56,141
40030 DATA248,139,189,3,192,24,105,12,15
7,14,192,173,21,208,29,231,111,141
40031 DATA21,208,169,4,157,10,192,234,16
9,16,234,234,234,173,16,208,61,223
40032 DATA111,141,16,208,189,14,192,10,1
53,254,207,144,9,173,16,208,29,231
40033 DATA111,141,16,208,76,11,116,76,20
4,112,0,0,189,12,192,41,17,240,244
40034 DATA189,16,192,208,239,224,2,208,2
2,189,3,192,56,233,12,157,14,192,169
40035 DATA59,141,251,139,169,60,141,249,
139,76,186,114,169,50,141,250,139,169
40036 DATA51,141,248,139,189,3,192,24,10
5,12,157,14,192,76,74,114,0,0,165,2
40037 DATA41,17,208,46,173,5,192,56,237,
4,192,201,18,144,32,201,22,176,28,173
40038 DATA250,139,201,49,240,18,201,58,2
40,14,173,251,139,201,49,240,7,201
40039 DATA58,240,3,76,24,115,76,61,115,7
6,234,112,173,5,192,56,237,4,192,201
40040 DATA15,144,242,201,19,176,238,173,
250,139,201,48,240,228,201,57,240,224
40041 DATA173,251,139,201,48,240,217,201
,57,240,213,169,0,141,4,212,141,6,212
40042 DATA169,8,141,5,212,169,4,141,1,21
2,165,2,41,18,208,5,169,5,141,1,212
40043 DATA169,129,141,4,212,76,48,116,16
9,0,141,4,212,141,6,212,169,3,141,5
40044 DATA212,169,4,141,1,212,165,2,41,1
8,208,5,169,5,141,1,212,169,129,141
40045 DATA4,212,76,234,112,0,0,254,22,19
2,152,72,189,22,192,41,3,224,2,240
40046 DATA24,201,0,208,3,254,24,192,188,
24,192,24,105,124,153,84,137,169,7
40047 DATA153,84,217,104,168,96,234,234,
201,0,208,3,222,24,192,188,24,192,24
40048 DATA105,120,153,51,137,169,2,153,5
1,217,104,168,96,234,234,0,0,238,22
40049 DATA192,173,22,192,201,5,208,81,16
9,0,141,22,192,206,21,192,16,18,169
40050 DATA9,141,21,192,206,20,192,16,8,1
69,5,141,20,192,206,19,192,169,18,133
40051 DATA211,169,7,133,214,32,108,229,1

```

<160>

<14>

<228>

<46>

<226>

<111>

<234>

<152>

<127>

<236>

<102>

<52>

<216>

<228>

<139>

<3>

<247>

<140>

<125>

<218>

<60>

<156>

<164>

<142>

<153>

<188>

<206>

<32>

<123>

<170>

<132>

<84>

<183>

<199>

<174>

<73>

<56>

<140>





# Commodore 64



```

69,30,32,210,255,173,19,192,32,4,116 <0>
40052 DATA169,58,32,210,255,173,20,192,3 <199>
2,4,116,173,21,192,32,4,116,173,19
40053 DATA192,24,109,20,192,109,21,192,2 <106>
08,2,104,104,96,24,105,48,76,210,255
40054 DATA0,254,28,192,189,28,192,201,3, <219>
208,8,169,0,157,28,192,222,26,192,169
40055 DATA50,56,253,26,192,74,74,24,105, <92>
8,157,16,192,76,191,114,0,0,32,100
40056 DATA115,165,2,41,17,208,17,173,5,1 <66>
92,56,237,4,192,201,20,208,3,32,100
40057 DATA115,76,109,116,173,5,192,56,23 <41>
7,4,192,201,17,208,3,32,100,115,76
40058 DATA109,116,0,0,162,2,160,4,32,61, <120>
113,162,1,160,2,32,61,113,96,0,189
40059 DATA22,192,74,141,33,192,32,198,11 <224>
6,56,237,33,192,16,6,169,255,157,33
40060 DATA192,96,189,22,192,201,54,176,2 <31>
43,76,234,112,0,0,120,169,49,141,20
40061 DATA3,169,234,141,21,3,88,96,0,0,1 <176>
62,0,189,0,96,157,0,134,189,192,96
40062 DATA157,192,134,232,224,192,208,23 <102>
9,96,162,0,189,128,97,157,0,134,189,64
40063 DATA98,157,192,134,232,224,192,208 <167>
,239,96
40070 DATA224,2,208,4,173,31,192,96,173, <184>
32,192,96
49985 REM <147>
49990 REM SPRITE-DATAS <243>
49995 REM <157>
50000 DATA0,0,0,0,0,0,0,0,0,0,0,0,85,0 <86>
,1,85,64,5,95,80,5,221,240,5,223,240
50005 DATA7,255,192,3,255,192,7,255,0,7, <65>
253,104,31,245,90,31,234,170,31,122
50010 DATA170,125,254,168,127,255,144,12 <24>
7,223,96,213,127,160,255,255,128,0
50100 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <193>
0,85,0,1,85,64,5,95,80,5,221,168
50105 DATA5,222,90,7,245,170,3,250,170,7 <94>
,250,168,7,222,168,61,247,240
50110 DATA61,255,208,63,127,64,255,127,6 <173>
4,255,213,0,255,255,0,255,255,0,0
50150 DATA0,0,0,0,0,0,0,0,0,0,0,0,85,0 <236>
,1,85,64,5,95,80,5,221,240,5,223,240
50155 DATA7,255,192,3,245,207,15,255,255 <193>
,63,255,255,63,127,255,63,213,168
50160 DATA255,255,168,255,255,168,255,25 <146>
5,160,255,255,0,255,255,0,255,255,0,0
50200 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <52>
0,0,0,0,0,0,0,2,164,0,10,90
50205 DATA15,250,170,255,250,168,255,202 <226>
,160,252,0,0,0,0,0,0,0,0,0,0,0,0,0
50210 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <105>
50240 DATA2,105,168,2,105,168,2,154,104, <15>
2,154,96,0,155,252
50245 DATA0,255,252,0,255,252,0,63,252,0 <129>
,63,240,0,255,240,0,253,64,3,245,0
50248 DATA1,85,64,1,69,84,5,69,85,5,84,0 <149>
,5,85,0,0,0,0,0,0,0,0,0,0,0,0,0
50250 DATA2,105,168,2,105,168,2,154,104, <25>
2,154,96,0,155,252
50255 DATA0,255,252,0,63,252,0,63,252,0, <202>
63,252,0,252,240,0,252,240,0,83,192
50260 DATA1,81,64,1,81,64,1,85,64,1,85,8 <188>
4,0,5,85,0,0,0,0,0,0,0,0,0,0,0,0
50300 DATA2,105,168,2,105,168,2,154,104, <49>
2,154,96,0,155,252,0,255,252,0,255,240
50305 DATA0,63,240,0,63,192,0,63,192,0,6 <95>
3,0,0,253,0,0,84,0,0,85,64
50310 DATA1,85,80,1,85,0,1,85,64,0,0,0,0, <128>
0,0,0,0,0,0,0,0,0,0,0,0,0
50350 DATA0,0,0,0,0,0,0,0,0,0,0,0,85,0 <181>
,1,85,64,5,95,80,5,221,240,5,223,240
50355 DATA5,255,192,3,255,192,15,255,0,6 <85>
3,255,0,63,247,160,63,253,168

```

```

50360 DATA255,255,255,253,255,255,255,95 <163>
,255,255,245,255,255,255,0,255,255,0,0
50400 DATA,,,,,,,,,,,,,,,,,,,,,,,,,,,,, <43>
,,,,,,,,,,,,,,,,0,10,144,255,41,104
50405 DATA255,234,168,255,234,160,0,10,1 <107>
28,0,0,0,0
51000 DATA0,0,0,0,0,0,0,0,0,0,0,0,85,0 <81>
,3,213,64,3,125,80,15,253,80
51005 DATA15,255,80,3,255,64,0,255,240,0, <238>
,255,240,41,127,252,165,93,252
51010 DATA170,169,255,170,175,125,42,191 <226>
,253,6,223,247,9,245,95,10,255,255
51015 DATA2,255,255,0 <95>
51050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <229>
0,85,0,3,213,64,42,125,80
51055 DATA169,189,80,170,95,80,170,175,6 <119>
4,42,175,240,42,191,240,15,223,220
51060 DATA3,247,220,1,255,124,0,127,124, <159>
,253,6,223,247,9,245,95,10,255,255
51065 DATA0,255,255,0 <141>
51100 DATA0,0,0,0,0,0,0,0,0,0,0,0,85,0 <79>
,3,213,64,3,125,80,15,253,80,15,255,80
51105 DATA3,255,64,0,255,240,252,215,240 <234>
,255,255,252,255,255,252,63,255,255
51110 DATA10,127,127,42,213,255,42,255,2 <209>
55,10,255,255,0,255,255,0,255,255,0
51150 DATA,,,,,,,,,,,,,,,,,,,,,,,,26,128 <1>
,0,165,160,0,170,175,252,42,175,255
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, <215>
51200 DATA41,166,128,41,166,128,41,166,1 <134>
28,57,166,128,63,230,0,63,255,0
51205 DATA63,255,0,63,255,0,15,252,0,15, <17>
255,0,1,127,0,0,95,192,1,85,64
51210 DATA21,81,64,85,81,80,0,21,80,0,85 <83>
,80,0,0,0,0,0,0,0,0,0,0,0,0,0,0
51250 DATA41,166,128,41,166,128,41,166,1 <202>
28,57,166,128,63,230,0,63,255,0,63,255
51255 DATA0,63,252,0,15,252,0,15,207,0,1 <1>
5,207,0,3,197,0,3,197,64,1,69,64
51260 DATA1,85,64,21,85,64,85,80,0,0,0,0, <49>
0,0,0,0,0,0,0,0,0,0,0,0,0
51300 DATA41,166,128,41,166,128,41,166,1 <234>
28,57,166,128,63,230,0,63,255,0
51305 DATA63,255,0,15,252,0,15,252,0,3,2 <163>
52,0,3,252,0,0,252,0,0,21,0
51310 DATA1,85,0,5,85,64,0,85,64,1,85,64 <230>
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
52000 DATA7,192,0,23,252,0,93,255,0,95,2 <117>
55,0,85,127,0,87,255,2,23,255,10
52005 DATA3,255,254,7,255,255,7,255,222, <29>
1,253,126,1,255,255,0,127,253,0,127,212
52010 DATA0,21,127,0,0,250,0,0,38,0,0,41 <251>
,0,0,42,0,0,10,0,0,2,0
52050 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <124>
128,0,0,160,0,0,80,0,0,169,0,0
52055 DATA150,128,0,170,128,0,170,0,0,17 <242>
2,0,0,255,0,0,234,0,0,154,128,0
52060 DATA166,160,0,169,160,0,107,240,0, <255>
183,252,0,253,255,0,0
52100 DATA255,127,0,63,127,192,15,223,19 <103>
2,3,223,0,15,207,0,63,5,0,63,5,64
52105 DATA84,5,84,85,5,85,85,80,0,21,84, <230>
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
52110 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <220>
52200 DATA0,0,0,0,0,0,0,0,1,0,0,23,21,80 <17>
,255,87,255,255,93,247,255,95,247,255
52205 DATA93,247,255,87,255,127,21,80,12 <92>
7,0,0,23,0,0,1,0,0,0,0,0,0,0,0,0
52210 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <20>
0
52250 DATA0,37,128,21,170,96,127,170,167 <198>
,255,170,91,245,245,171,255,250,171
52255 DATA255,250,171,255,250,171,215,24 <2>
5,171,253,122,91,255,220,165,255,246

```





```

52755 DATA253,255,20,247,255,0,247,255,0
,15,240,0,0,2,0,0,2,0,0,0,0,0,0,0,0    <59>
52760 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0                                          <73>
53000 DATA0,0,0,0,0,0,0,0,0,0,0,0,85,0
,3,213,64,3,125,80,15,253,80              <41>
53005 DATA15,255,80,3,255,64,0,255,240,0
,255,240,0,255,252,42,223,252            <108>
53010 DATA170,127,255,255,255,255,255,25
5,127,255,245,255,192,95,255             <20>
53015 DATA0,255,255,0,255,255,0          <213>
53050 DATA,,,,,,,,,,,,,,
,,,,,,,6,160,0,41,104,255                <232>
53055 DATA42,171,255,10,171,255,2,160,0,
0,0,0,0                                     <101>
54985 REM                                   <46>
54770 REM                                ZEICHENDATAS <247>
54995 REM                                   <56>
55000 DATA65,87,87,93,93,117,117,213,213 <253>
55005 DATA66,213,213,117,117,93,93,87,87 <17>
55010 DATA67,87,87,95,95,127,127,255,255 <191>
55015 DATA68,213,213,245,245,253,253,255
,255                                         <196>
55020 DATA69,235,235,235,235,235,235,235
,235                                         <116>
55025 DATA70,85,245,255,255,253,245,213,
85                                           <139>
55030 DATA71,85,87,95,95,127,127,255,255 <183>
55035 DATA72,85,213,245,245,253,253,255,
255                                         <45>
55040 DATA73,85,95,255,255,127,95,87,85 <162>
55045 DATA74,85,85,255,255,85,85,85,85 <200>
55055 DATA75,3,15,15,63,63,255,255,255 <25>
55060 DATA76,192,240,240,252,252,255,255
,255                                         <188>
55065 DATA77,255,255,255,252,252,248,248
,192                                         <212>
55070 DATA78,255,255,255,63,63,15,15,3   <49>
55075 DATA79,255,255,235,235,235,235,235
,235                                         <206>

```



**Retten Sie Ihr Raumschiff vor den Weltraumungeheuern! Ein tolles Maschinen-  
spracheprogramm für alle, die sich von „Monstern“ nicht beeindrucken lassen!**

## Hinweise zum Programm

Zuerst muß das Programm „Monster-Lader“ geladen werden. Wenn das Programm gestartet wird, werden Basicstart und -ende geändert, um die Spritedaten zu sichern. Dann wird das Hauptprogramm nachgeladen und gestartet.

Nach dem Programmstart wird zunächst angezeigt, daß das Hauptprogramm geladen wird. Danach erscheinen auf dem Bildschirm die Monster, die abzuschießen sind. Es wird Punktzahl, Level, Punktzahl für Bonusschiff und die Belegung der F-Tasten angezeigt. Die Musik wird eingeschaltet.

(F1-Taste) -

Das Spiel wird gestartet.

(F7-Taste) -

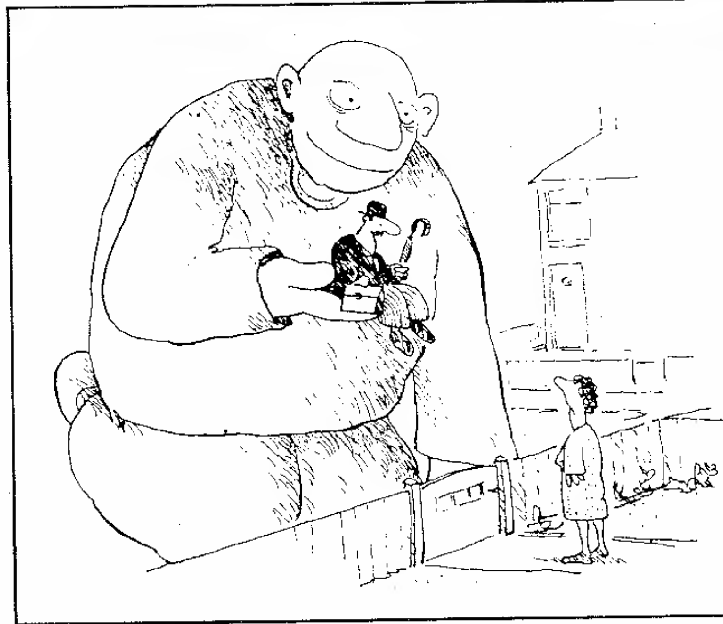
Die Spielstärke wird verändert.

(F5-Taste) -

Die fünf besten Ergebnisse werden angezeigt. Durch Drücken der „Return“-Taste werden die „High-Scores“ neu abgespeichert. Anschließend wird überprüft und angezeigt, ob beim Speichern ein Fehler aufgetreten ist.

Nach Drücken der F1-Taste beginnt das Spiel. Es gilt die verschiedenen Monster abzuschießen, bevor das eigene Raumschiff getroffen wird. Es werden „High-Score“, „Score“ und Anzahl der Raumschiffe angezeigt. Drei Ersatzschiffe stehen zur Verfügung. Die Anzahl der angreifenden Monster wird bei steigender Punktzahl erhöht. Bei Erreichen von 4000 Punk-

# MONSTERJAGD



**Lassen Sie  
sich während  
Ihrer Monster-  
jagd bloß nicht  
auf den Arm  
nehmen!**

ten erhält der Spieler ein zusätzliches Raumschiff. Sind alle Raumschiffe vernichtet, ist das Spiel zu Ende. Ein eventuelles Bestergebnis wird angezeigt. Gehört die erreichte Punktzahl zu den „Besten

Fünf“, kann nach Ende des Spiels der Name des Spielers eingegeben werden. Er wird dann in die Bestenliste aufgenommen. Durch Tastendruck wird das Programm neu gestartet.

## Das Listing

### Teil Eins

```
20 POKE53280,7:POKE53281,7 <21>
30 PRINT" (CLEAR BLACK) LOADING (SPACE) MONST
ERJAGD....." <22>
35 POKE45,28:POKE46,81:POKE47,28 <20>
37 POKE48,81:POKE49,28:POKE50,81 <195>
40 POKE43,160:POKE44,15:POKE3999,0:CLR <75>
50 LOAD"MONSTERJAGD",8 <97>
```

### Hauptprogramm

```
100 REM ***** <205>
102 REM ***** MONSTERJAGD <203>
104 REM ***** VON <240>
106 REM ***** GERHARD STOEHR <126>
108 REM ***** (C) 1985 <251>
110 REM ***** <215>
112 REM ***** <255>
114 REM <1>
116 GOSUB444:REM *** SCORES LADEN *** <197>
118 REM <5>
120 REM **** DATAS EINLESEN **** <26>
```

```
122 FORS=832 TO894 :READQ:POKES,Q:NEXT <63>
124 FORS=896 TO958 :READQ:POKES,Q:NEXT <131>
126 FORS=960 TO1022:READQ:POKES,Q:NEXT <66>
128 FORS=2048TO2110:READQ:POKES,Q:NEXT <114>
130 FORS=2112TO2174:READQ:POKES,Q:NEXT <175>
132 FORS=2176TO2238:READQ:POKES,Q:NEXT <255>
134 FORS=2240TO2302:READQ:POKES,Q:NEXT <107>
136 FORS=2304TO2366:READQ:POKES,Q:NEXT <232>
138 FORS=2368TO2430:READQ:POKES,Q:NEXT <203>
140 FORS=2432TO2494:READQ:POKES,Q:NEXT <8>
142 FORS=2496TO2558:READQ:POKES,Q:NEXT <88>
144 FORS=2560TO2622:READQ:POKES,Q:NEXT <196>
146 FORS=2624TO2686:READQ:POKES,Q:NEXT <65>
148 FORS=2688TO2750:READQ:POKES,Q:NEXT <36>
150 FORS=2752TO2814:READQ:POKES,Q:NEXT <253>
152 FORS=2816TO2878:READQ:POKES,Q:NEXT <122>
154 FORS=2880TO2942:READQ:POKES,Q:NEXT <29>
156 DINS(19):FORI=1TO19:READS(1):NEXT <86>
158 FORI=49152TO50439:READQ:POKEI,Q <174>
159 NEXT <33>
160 REM **** DATAS EINLESEN **** <66>
162 REM <49>
164 REM **** SPRITE POINTER **** <97>
166 POKE2040,13:POKE2041,14 <168>
```





# Commodore 64



```

168 POKE2042,15:POKE2043,15 <241>
170 POKE2044,34:POKE2045,36 <80>
172 POKE2046,40:POKE2047,42 <43>
174 REM **** SPRITE POINTER **** <107>
176 REM <63>
178 AS=B(1) :REM **** HIGH SCORE <57>
180 L=1 :REM **** LEVEL <54>
182 V=53248 :REM **** V DEFINIEREN <118>
184 POKE5328,0 :REM **** UHR AN <66>
186 REM <73>
188 POKEV+40,1:POKEV+44,0:POKEV+44,8 <166>
190 POKEV+46,2: REM **** SPRITE FARBEN <189>
192 REM <79>
194 REM **** X-Y KOORDINATEN **** <14>
196 POKEV+0,0 :POKEV+1,0 <84>
198 POKEV+2,0 :POKEV+3,0 <116>
200 POKEV+4,170 :POKEV+5,80 <215>
202 POKEV+8,170 :POKEV+9,152 <86>
204 POKEV+10,170:POKEV+11,129 <64>
206 POKEV+12,170:POKEV+13,106 <13>
208 REM **** X-Y KOORDINATEN **** <28>
210 REM <97>
212 POKE53280,0:POKE53281,0:REM****FARBE <140>
214 POKEV+21,253:REM **** SPRITE EIN <167>
216 SYS49201 :REM **** MUSIK AN <172>
218 REM <105>
220 REM **** VORSPANN **** <20>
222 PRINT"(CLEAR YELLOW SPACE)MONSTERJAG <22>
D";
224 PRINT"(SPACE5)VON(SPACE2)GERHARD(SPA <177>
CE2)STOEHR"
226 PRINT"(YELLOW SPACE) (SPAC <43>
ES) (SHIFTSPACE SPACE) (SHIFTSPA
CE SPACE)"
228 PRINT"(DOWN3 CYAN SPACE)100(SPACE)PO <25>
INTS"
230 PRINT"(DOWN2 YELLOW SPACE)100(SPACE) <82>
POINTS"
232 PRINT"(DOWN2 ORANGE SPACE)200(SPACE) <202>
POINTS"
234 PRINT"(DOWN2 GREEN SPACE)000(SPACE)P <207>
OINTS"
236 PRINT"(DOWN3 WHITE SPACE)LEVEL(SPACE <244>
)";L
238 PRINT"(DOWN2 LIG.RED SPACE)AT(SPACE) <54>
4000(SPACE)POINTS(SPACE)BONUS(SPACE)SHIP
"
240 PRINT"(DOWN2 CYAN SPACE)F1(SPACE)PLA <214>
Y(SPACE)GAME(SPACE2)F5(SPACE)HIGH(SPACE)
SCORE(SPACE2)F7(SPACE)LEVEL(UP6)"
242 GET X$:IFX#=""THEN250 <70>
244 IFASC(X$)=133THEN268 <15>
246 IFASC(X$)=135THEN396 <60>
248 IFASC(X$)=136THEN258 <36>
250 Z=Z+1:IFZ=20THENPOKE2042,32:POKE2044 <177>
,34:POKE2045,36:POKE2046,40
252 IFZ=10THENPOKE2042,15:POKE2044,35:PO <178>
KE2045,37:POKE2046,41
254 IFZ=20THENZ=0 <171>
256 GOTO242 <82>
258 L=L+1:IFL=9THENL=1 <174>
260 PRINT"(UP RIGHT7 WHITE)";L:GOTO242 <190>
262 REM **** VORSPANN **** <63>
264 REM <152>
266 REM **** X-Y KOORDINATEN **** <87>
268 POKEV+0,135:POKEV+1,220 <218>
270 POKEV+2,0 :POKEV+3,125 <161>
272 POKEV+4,0 :POKEV+5,0 <221>
274 POKEV+6,0 :POKEV+7,0 <253>
276 POKEV+8,0 :POKEV+9,220 <195>
278 POKEV+10,0 :POKEV+11,70 <151>
280 POKEV+12,0 :POKEV+13,190 <35>
282 POKEV+14,0 :POKEV+15,220 <232>
284 REM **** X-Y KOORDINATEN **** <105>
286 REM <174>
288 POKE53280,6 : REM **** FARBE <210>

```

```

290 T=0 : REM **** HIGH SCORE <172>
292 POKEV+30,0 : REM **** KOLLISION <146>
294 POKE2040,13 : REM **** POINTER <140>
296 POKEV+23,0:POKEV+29,0:REM ** KLEIN <221>
298 POKEV+39,1 : REM **** FARBE1 <185>
300 REM <188>
302 REM **** ANZEIGE WOERTER **** <154>
304 PRINT"(CLEAR YELLOW DOWN3 LEFT6)SHIP <174>
S";
306 PRINT"(DOWN5 LEFT5)SCORE"; <95>
308 PRINT"(DOWN5 LEFT5)HIGH(DOWN LEFT4)S <212>
CORE"
310 PRINT"(WHITE DOWN2 LEFT7)"AS*100 <117>
312 REM **** ANZEIGE WOERTER **** <164>
314 REM <202>
316 REM **** ANZEIGE ZAHLEN **** <189>
318 POKE1220,51:POKE55492,1 <237>
320 POKE1418,48:POKE1419,48:POKE1420,48: <173>
POKE1421,48:POKE1422,48
322 POKE55690,1:POKE55691,1:POKE55692,1: <24>
POKE55693,1:POKE55694,1
324 REM **** ANZEIGE ZAHLEN **** <197>
326 REM <214>
328 FORD=1056T02016STEP40:POKE1,160 <139>
330 POKE1+54272,6:NEXT:REM****BILDTEIL. <77>
332 REM <220>
334 FORI=1TO19:POKE5(I),42 <79>
336 POKE5(I)+54272,6:NEXT:REM****STERNE <175>
338 REM <226>
340 POKE780,9-L :REM **** SET LEVEL <246>
342 SYS49152 :REM **** MASCHINENPR. <131>
344 REM <232>
346 REM **** HIGH SCORE **** <60>
348 NS=PEEK(780):IFNS>ASTHENAS=NS:T=1 <119>
350 IFT=1THENPRINT"(HOME YELLOW DOWN2 RI <5>
GHT2)NEW(SPACE)HIGH(SPACE)SCORE";:FORW=1
TO90:NEXT <32>
352 IFT=1THENPRINT"(RED LEFT14)NEW(SPACE <68>
)HIGH(SPACE)SCORE":FORW=1TO90:NEXT
354 REM **** HIGH SCORE **** <244>
356 REM <10>
358 GETA$:IFA#="(F1)"THEN366 <151>
360 GOTO350: REM **** NOCH MAL **** <250>
362 REM <215>
364 REM **** BEST FIVE **** <158>
366 IFNS=<B(5) THEN196 <89>
368 POKE53280,6:POKE53281,6
370 POKEV+21,0:PRINT"(CLEAR WHITE DOWNS <183>
SPACE3)WRITE(SPACE)YOUR(SPACE)NAME(SPACE
)AND(SPACE)PRESS(SPACE)RETURN(DOWN4 RIGH
T8 BLACK)";
372 IFNS>B(1) THENGOSUB382:GOSUB384:GOSUB <200>
386:GOSUB388:INPUTB$(1):GOTO196
374 IFNS>B(2) THENGOSUB382:GOSUB384:GOSUB <184>
386:B(2)=NS:INPUTB$(2):GOTO196
376 IFNS>B(3) THENGOSUB382:GOSUB384:B(3)= <251>
NS:INPUTB$(3):GOTO196
378 IFNS>B(4) THENGOSUB382:B(4)=NS:INPUTB <86>
$(4):GOTO196
380 IFNS>B(5) THENB(5)=NS:INPUTB$(5):GOTO <181>
196
382 B(5)=B(4):B$(5)=B$(4):RETURN <207>
384 B(4)=B(3):B$(4)=B$(3):RETURN <164>
386 B(3)=B(2):B$(3)=B$(2):RETURN <121>
388 B(2)=B(1):B$(2)=B$(1):B(1)=NS:RETURN <224>
390 REM **** BEST FIVE **** <241>
392 REM <24>
394 REM **** BEST FIVE ANZEIGE **** <173>
396 POKEV+21,0:POKE53280,1:POKE53281,1 <86>
398 PRINT"(CLEAR BLACK DOWN2 SPACE13)BES <119>
T(SPACE)FIVE"
400 PRINT"(SPACE13) (DOWN3)" <120>
402 FORI=1TO5 <242>
404 PRINT"(DOWN RED RIGHT)";B$(I); <1>
406 PRINTTAB(19)"(BLUE)";B(I)*100; <236>

```



```

408 PRINTTAB(28)"(GREEN)POINTS"
410 NEXT
412 PRINT"(DOWN4 RIGHT2 BLACK)PRESS(SPAC
E)RETURN(SPAC)TD(SPAC)SAVE(SPAC)HIGH
(SPAC)SCORES"
414 GETZ$:IFZ$=""THEN414
416 IFASC(Z$)<>13THENGOTO196
418 GOSUB430:GOSUB456
420 IFFE$="OK"THENPRINT"(DOWN SPACE2)OK!
":GOTO414
422 PRINT"(DOWN SPACE2)DISK(SPAC)ERROR(
SPACE)!!":GOTO414
424 REM **** BEST GIVE ANZEIGE ****
426 REM
428 REM **** SCORES SPEICHERN ****
430 OPEN1,8,2,"@:MONSTER(SPAC)SCORES,S,
W"
432 FORI=1TO5
434 PRINT#1,B(I):PRINT#1,B$(I)
436 NEXT:CLOSE1:RETURN
438 REM **** SCORES SPEICHERN ****
440 REM
442 REM **** SCORES LADEN ****
444 OPEN1,8,2,"MONSTER(SPAC)SCORES"
446 FORI=1TO5
448 INPUT#1,B(I):INPUT#1,B$(I)
450 NEXT:CLOSE1:RETURN
452 REM **** SCORES LADEN ****
454 REM
456 OPEN1,8,15:INPUT#1,FE,FE$,FA,FB
458 CLOSE1:RETURN:REM *** DISK FEHLER
460 REM
462 REM
464 REM **** DATA FLUGZEUG ****
466 DATA0,0,0,0,0,0
468 DATA0,0,0,0,28,0
470 DATA0,28,0,0,62,0
472 DATA0,127,0,0,62,0
474 DATA0,28,0,6,127,48
476 DATA6,127,48,6,255,176
478 DATA7,156,240,103,62,115
480 DATA102,127,51,108,255,155
482 DATA121,255,207,115,255,231
484 DATA127,255,255,3,193,224,3,193,224
486 REM **** DATA FLUGZEUG ****
488 REM **** DATA FEUER ****
490 DATA0,28,0,0,28,0
492 DATA0,28,0,0,8,0
494 DATA0,0,0,0,0,0
496 DATA0,0,0,0,0,0
498 DATA0,0,0,0,0,0
500 DATA0,0,0,0,0,0
502 DATA0,0,0,0,0,0
504 DATA0,0,0,0,0,0
506 DATA0,0,0,0,0,0
508 DATA0,0,0,0,0,0
510 DATA0,0,0
512 REM **** DATA FEUER ****
514 REM **** DATA TOTENKOPF 1 ****
516 DATA0,0,0,24,0,24
518 DATA24,0,24,48,0,12
520 DATA248,60,31,220,255,59
522 DATA15,255,240,7,255,224
524 DATA3,153,192,7,24,224
526 DATA7,24,224,7,255,224
528 DATA3,231,192,7,231,224
530 DATA14,255,112,220,0,59
532 DATA248,0,31,48,231,12
534 DATA24,126,56,24,60,48,0,0,0
536 REM **** DATA TOTENKOPF 1 ****
538 REM **** DATA TOTENKOPF 2 ****
540 DATA0,0,0,24,0,24
542 DATA24,0,24,48,0,12
544 DATA248,60,31,220,255,59
546 DATA15,255,240,7,255,224
548 DATA3,153,192,7,24,224

```

```

<215>
<29>
<235>
<31>
<196>
<91>
<137>
<143>
<213>
<58>
<177>
<248>
<16>
<191>
<197>
<187>
<72>
<158>
<166>
<30>
<5>
<211>
<168>
<86>
<208>
<248>
<92>
<94>
<85>
<58>
<20>
<196>
<150>
<170>
<170>
<117>
<229>
<26>
<104>
<107>
<140>
<244>
<60>
<86>
<88>
<90>
<92>
<94>
<96>
<98>
<100>
<34>
<165>
<217>
<251>
<20>
<96>
<37>
<25>
<61>
<208>
<170>
<167>
<181>
<239>
<4>
<19>
<44>
<120>
<61>
<49>

```

# MONSTER JAGD

```

550 DATA7,24,224,7,255,224
552 DATA3,231,192,7,231,224
554 DATA14,255,112,220,255,59
556 DATA248,60,31,48,0,12
558 DATA24,0,56,24,0,48,0,0,0
560 REM **** DATA TOTENKOPF 2 ****
562 REM **** DATA KREUZ ****
564 DATA0,24,0,0,24,0
566 DATA0,24,0,56,0,28
568 DATA28,126,56,14,102,112
570 DATA4,102,32,0,102,0
572 DATA15,231,240,236,0,55
574 DATA236,0,55,15,231,240
576 DATA0,102,0,2,102,64
578 DATA7,102,224,14,102,112
580 DATA28,102,56,0,102,0
582 DATA0,102,0,0,102,0,0,126,0
584 REM **** DATA KREUZ ****
586 REM **** DATA SPINNE1 ****
588 DATA0,0,0,0,0,0
590 DATA0,0,0,0,0,0
592 DATA0,0,0,0,0,0
594 DATA0,0,0,0,48,0
596 DATA0,104,0,0,201,224
598 DATA57,139,48,47,250,28
600 DATA99,254,6,71,255,194
602 DATA67,252,98,70,48,18
604 DATA76,24,18,200,12,18
606 DATAB,4,16,8,4,0,0,4,0
608 REM **** DATA SPINNE1 ****
610 REM **** DATA SPINNE2 ****
612 DATA0,0,0,0,0,0
614 DATA0,0,0,0,0,0
616 DATA0,0,0,0,0,0
618 DATA0,0,0,0,112,0
620 DATA120,200,0,200,9,240
622 DATA136,11,28,143,250,6
624 DATA135,254,2,143,255,194
626 DATA15,252,96,6,48,16
628 DATA2,24,8,2,14,4
630 DATA2,1,128,2,0,128,0,0,128
632 REM **** DATA SPINNE2 ****
634 REM **** DATA VOGEL 1 ****
636 DATA240,0,0,126,0,0
638 DATA63,0,96,63,192,240
640 DATA31,224,248,15,225,239
642 DATA31,241,224,63,243,240
644 DATA15,255,152,3,255,0
646 DATA15,255,128,31,255,128
648 DATA7,255,0,1,254,0
650 DATA0,120,0,1,236,0
652 DATA63,199,192,7,131,128
654 DATA3,1,0,2,0,0,0,0,0
656 REM **** DATA VOGEL 1 ****
658 REM **** DATA VOGEL 2 ****
660 DATA0,0,0,0,0,0
662 DATA0,0,224,0,1,240
664 DATA0,1,184,0,195,252
666 DATA15,225,255,255,243,224
668 DATA63,255,192,15,255,0
670 DATA1,255,128,1,255,128
672 DATA15,255,0,3,254,0
674 DATA0,120,0,0,236,0
676 DATA3,198,0,31,131,192
678 DATA3,1,128,6,0,128,0,0,0
680 REM **** DATA VOGEL 2 ****
682 REM **** DATA HAEHNCHEN 1 ****
684 DATA0,0,0,0,6,0
686 DATA4,2,0,12,6,0
688 DATAB,196,0,12,64,130
690 DATA4,120,242,39,12,19
692 DATA97,132,49,64,220,33
694 DATA113,145,227,19,16,130
696 DATA48,0,134,32,240,0
698 DATA35,252,6,7,254,254
700 DATA14,127,230,221,191,192

```

```

<85>
<232>
<193>
<255>
<186>
<26>
<24>
<251>
<128>
<131>
<236>
<216>
<173>
<129>
<61>
<155>
<53>
<46>
<2>
<181>
<183>
<185>
<167>
<209>
<215>
<1>
<41>
<2>
<3>
<24>
<42>
<205>
<207>
<209>
<8>
<105>
<36>
<45>
<80>
<182>
<163>
<64>
<184>
<5>
<171>
<70>
<113>
<48>
<38>
<145>
<86>
<231>
<205>
<206>
<223>
<253>
<101>
<19>
<132>
<100>
<205>
<246>
<100>
<140>
<41>
<245>
<207>
<81>
<235>
<171>
<186>
<134>
<193>
<73>
<25>
<117>

```





# Commodore 64



```

702 DATA223,223,227,7,239,127,1,252,3
704 REM **** DATA HAEHNCHEN 1 ****
706 REM **** DATA HAEHNCHEN 2 ****
708 DATA4,0,0,68,6,8
710 DATA92,34,72,80,38,200
712 DATA211,100,142,146,65,2
714 DATA27,121,178,9,12,147
716 DATA193,128,177,128,192,33
718 DATA241,140,99,19,7,14
720 DATA48,1,24,32,241,0
722 DATA35,252,6,7,158,254
724 DATA14,111,230,221,247,192
726 DATA223,251,227,7,255,127,1,252,3
728 REM **** DATA HAEHNCHEN 2 ****
730 REM **** DATA SCHLANGE 1 ****
732 DATA0,0,0,0,0,0
734 DATA0,0,0,0,0,0
736 DATA0,0,0,0,8,0
738 DATA0,60,0,28,126,28
740 DATA126,119,62,247,227,179
742 DATA227,193,255,1,128,238
744 DATA0,0,0,0,0,0
746 DATA0,0,0,0,0,0
748 DATA0,0,0,0,0,0
750 DATA0,0,0,0,0,0,0,0
752 REM **** DATA SCHLANGE 1 ****
754 REM **** DATA SCHLANGE 2 ****
756 DATA0,0,0,0,0,0
758 DATA0,0,0,0,0,0
760 DATA0,0,0,0,0,0
762 DATA0,0,28,0,0,62
764 DATA28,28,127,63,127,127
766 DATA247,247,249,193,192,236
768 DATA0,0,6,0,0,0
770 DATA0,0,0,0,0,0
772 DATA0,0,0,0,0,0
774 DATA0,0,0,0,0,0,0,0
776 REM **** DATA SCHLANGE 2 ****
778 REM **** DATA BAGGER 1 ****
780 DATA160,0,0,80,0,0
782 DATA168,0,0,67,252,0
784 DATA35,252,0,63,28,0
786 DATA3,28,1,3,28,3
788 DATA3,28,7,255,31,206
790 DATA248,31,206,248,31,204
792 DATA248,31,252,255,255,252
794 DATA255,255,222,0,0,14
796 DATA51,51,39,127,255,243
798 DATA230,103,51,127,255,241,51,51,32
800 REM **** DATA BAGGER 1 ****
802 REM **** DATA BAGGER 2 ****
804 DATA0,0,0,0,0,0
806 DATA0,0,0,51,252,0
808 DATA51,252,0,63,28,0
810 DATA3,28,1,3,28,3
812 DATA3,28,7,255,31,206
814 DATA248,31,206,248,31,204
816 DATA248,31,252,255,255,252
818 DATA255,255,222,0,0,14
820 DATA102,102,39,255,255,243
822 DATA76,206,51,255,255,241
824 DATA89,153,32
826 REM **** DATA BAGGER 2 ****
828 REM **** DATA MUELL ****
830 DATA0,0,0,0,0,0
832 DATA0,0,0,0,0,0
834 DATA0,0,0,0,0,0
836 DATA0,12,0,0,58,0
838 DATA1,215,0,3,85,192
840 DATA0,102,64,1,223,96
842 DATA6,201,176,24,158,212
844 DATA15,184,140,3,101,178
846 DATA7,222,100,30,107,204
848 DATA60,200,126,0,96,51,0,0,0
850 REM **** DATA MUELL ****
852 REM **** DATA GAME OVER ****

```

```

<117>
<229>
<250>
<191>
<152>
<107>
<9>
<29>
<39>
<1>
<96>
<134>
<108>
<16>
<101>
<69>
<71>
<153>
<215>
<204>
<168>
<81>
<83>
<85>
<19>
<123>
<143>
<93>
<95>
<97>
<21>
<98>
<114>
<142>
<108>
<110>
<44>
<166>
<70>
<138>
<139>
<143>
<112>
<109>
<24>
<41>
<66>
<121>
<131>
<92>
<110>
<142>
<253>
<159>
<136>
<133>
<48>
<65>
<90>
<64>
<163>
<64>
<134>
<31>
<168>
<170>
<172>
<79>
<73>
<197>
<80>
<51>
<120>
<123>
<53>
<18>

```

```

854 DATA120,136,47,193,76,104
856 DATA130,42,168,130,41,46
858 DATA187,232,40,138,40,40
860 DATA202,40,40,122,40,47
862 DATA0,0,0,0,0,0
864 DATA114,47,121,138,40,69
866 DATA138,40,69,138,46,121
868 DATA138,40,81,138,40,73
870 DATA137,72,68,112,143,69
872 DATA0,0,0,0,0,0,0,0
874 REM **** DATA GAME OVER ****
876 REM **** DATA STERNE ****
878 DATA1065,1198,1610,1408,1524,1274
880 DATA1321,1225,1429,1553,1750,1933
882 DATA1089,1212,1075,1666,1886,1720
884 DATA1909
886 REM **** DATA STERNE ****
888 REM **** DATA MASCHINENPROGRAMM ***
890 DATA 32, 56, 192, 32, 199, 195
892 DATA 32, 242, 196, 32, 64, 192
894 DATA 32, 174, 192, 32, 238, 193
896 DATA 32, 207, 192, 32, 1, 193
898 DATA 32, 82, 194, 32, 180, 194
900 DATA 32, 208, 194, 32, 236, 194
902 DATA 32, 13, 193, 173, 196, 4
904 DATA 201, 47, 240, 3, 76, 9
906 DATA 192, 32, 246, 195, 173, 237
908 DATA 195, 96, 160, 0, 140, 237
910 DATA 195, 141, 243, 195, 173, 1
912 DATA 220, 74, 176, 3, 32, 131
914 DATA 192, 74, 176, 3, 32, 120
916 DATA 192, 74, 176, 3, 32, 109
918 DATA 192, 74, 176, 3, 32, 98
920 DATA 192, 74, 176, 3, 32, 142
922 DATA 192, 96, 238, 0, 208, 174
924 DATA 0, 208, 224, 255, 240, 1
926 DATA 96, 206, 0, 208, 174, 0
928 DATA 208, 224, 24, 240, 235, 96
930 DATA 238, 1, 208, 174, 1, 208
932 DATA 224, 222, 240, 1, 96, 206
934 DATA 1, 208, 174, 1, 208, 224
936 DATA 190, 240, 235, 96, 173, 238
938 DATA 195, 208, 26, 173, 0, 208
940 DATA 141, 2, 208, 173, 1, 208
942 DATA 56, 233, 4, 141, 3, 208
944 DATA 173, 21, 208, 9, 2, 141
946 DATA 21, 208, 238, 238, 195, 96
948 DATA 173, 238, 195, 240, 27, 206
950 DATA 3, 208, 206, 3, 208, 206
952 DATA 3, 208, 173, 3, 208, 201
954 DATA 60, 176, 11, 206, 238, 195
956 DATA 173, 21, 208, 41, 253, 141
958 DATA 21, 208, 96, 165, 162, 69
960 DATA 161, 13, 9, 220, 141, 241
962 DATA 195, 201, 120, 144, 8, 201
964 DATA 230, 144, 18, 238, 5, 208
966 DATA 96, 206, 4, 208, 238, 5
968 DATA 208, 173, 4, 208, 201, 20
970 DATA 240, 1, 96, 238, 4, 208
972 DATA 238, 5, 208, 173, 4, 208
974 DATA 201, 235, 240, 229, 96, 174
976 DATA 243, 195, 160, 255, 136, 208
978 DATA 253, 202, 208, 248, 96, 173
980 DATA 30, 208, 160, 0, 140, 30
982 DATA 208, 201, 129, 240, 43, 201
984 DATA 3, 240, 39, 74, 144, 4
986 DATA 32, 69, 193, 96, 74, 144
988 DATA 29, 74, 144, 4, 32, 148
990 DATA 193, 96, 74, 144, 4, 32
992 DATA 208, 193, 96, 74, 74, 144
994 DATA 4, 32, 28, 195, 96, 74
996 DATA 144, 4, 32, 178, 193, 96
998 DATA 96, 169, 0, 141, 10, 208
1000 DATA 141, 8, 208, 141, 5, 208
1002 DATA 141, 7, 208, 141, 12, 208
1004 DATA 141, 242, 195, 169, 44, 141
1006 DATA 248, 7, 206, 176, 4, 32

```



## LISTING : The End

1008 DATA 115, 193, 169, 129, 141, 21	<161>	1166 DATA 193, 32, 82, 194, 173, 1	<140>
1010 DATA 208, 32, 185, 195, 32, 56	<139>	1168 DATA 208, 141, 15, 208, 238, 14	<221>
1012 DATA 195, 32, 199, 195, 96, 169	<20>	1170 DATA 208, 173, 14, 208, 24, 105	<140>
1014 DATA 15, 141, 24, 212, 169, 240	<28>	1172 DATA 22, 205, 0, 208, 208, 228	<180>
1016 DATA 141, 6, 212, 169, 33, 141	<212>	1174 DATA 238, 0, 208, 173, 0, 208	<83>
1018 DATA 4, 212, 169, 255, 141, 1	<166>	1176 DATA 208, 220, 169, 13, 141, 248	<138>
1020 DATA 212, 162, 0, 232, 208, 253	<3>	1178 DATA 7, 173, 196, 4, 201, 47	<202>
1022 DATA 233, 1, 208, 244, 141, 1	<77>	1180 DATA 208, 28, 169, 45, 141, 248	<147>
1024 DATA 212, 96, 162, 33, 142, 250	<255>	1182 DATA 7, 169, 1, 141, 23, 208	<136>
1026 DATA 7, 173, 21, 208, 41, 253	<214>	1184 DATA 141, 29, 208, 169, 5, 141	<215>
1028 DATA 141, 21, 208, 32, 115, 193	<63>	1186 DATA 39, 208, 169, 220, 141, 15	<169>
1030 DATA 32, 33, 194, 162, 15, 142	<231>	1188 DATA 208, 169, 204, 141, 1, 208	<147>
1032 DATA 250, 7, 162, 0, 142, 5	<238>	1190 DATA 162, 5, 32, 4, 193, 32	<228>
1034 DATA 208, 96, 162, 33, 142, 254	<118>	1192 DATA 82, 194, 238, 14, 208, 238	<136>
1036 DATA 7, 173, 21, 208, 41, 253	<224>	1194 DATA 0, 208, 173, 0, 208, 201	<6>
1038 DATA 141, 21, 208, 32, 115, 193	<73>	1196 DATA 135, 208, 235, 162, 5, 32	<123>
1040 DATA 32, 33, 194, 162, 40, 142	<204>	1198 DATA 4, 193, 32, 82, 194, 169	<143>
1042 DATA 254, 7, 162, 0, 142, 12	<118>	1200 DATA 0, 141, 30, 208, 206, 14	<50>
1044 DATA 208, 96, 162, 33, 142, 251	<62>	1202 DATA 208, 173, 14, 208, 208, 235	<165>
1046 DATA 7, 173, 21, 208, 41, 253	<234>	1204 DATA 169, 253, 141, 21, 208, 169	<221>
1048 DATA 141, 21, 208, 32, 115, 193	<83>	1206 DATA 0, 141, 30, 208, 96, 162	<232>
1050 DATA 32, 33, 194, 162, 15, 142	<251>	1208 DATA 0, 189, 212, 195, 157, 0	<204>
1052 DATA 251, 7, 162, 0, 142, 7	<42>	1210 DATA 212, 232, 224, 25, 208, 245	<133>
1054 DATA 208, 96, 173, 237, 195, 201	<53>	1212 DATA 96, 169, 0, 162, 0, 157	<204>
1056 DATA 5, 176, 1, 96, 173, 241	<109>	1214 DATA 0, 212, 232, 224, 25, 208	<146>
1058 DATA 195, 201, 120, 144, 22, 201	<179>	1216 DATA 248, 96, 9, 2, 0, 3	<247>
1060 DATA 230, 144, 4, 238, 7, 208	<81>	1218 DATA 65, 0, 240, 12, 2, 0	<97>
1062 DATA 96, 206, 6, 208, 238, 7	<172>	1220 DATA 4, 65, 0, 192, 16, 2	<30>
1064 DATA 208, 173, 6, 208, 201, 20	<116>	1222 DATA 0, 6, 65, 0, 64, 0	<2>
1066 DATA 240, 1, 96, 238, 6, 208	<155>	1224 DATA 30, 243, 31, 0, 0, 0	<53>
1068 DATA 238, 7, 208, 173, 6, 208	<115>	1226 DATA 0, 0, 0, 0, 0, 0	<53>
1070 DATA 201, 235, 240, 229, 96, 238	<216>	1228 DATA 120, 169, 3, 160, 196, 141	<50>
1072 DATA 237, 195, 173, 237, 195, 201	<242>	1230 DATA 20, 3, 140, 21, 3, 88	<163>
1074 DATA 40, 208, 3, 238, 196, 4	<169>	1232 DATA 96, 169, 10, 141, 24, 212	<66>
1076 DATA 238, 140, 5, 173, 140, 5	<201>	1234 DATA 169, 240, 141, 20, 212, 141	<137>
1078 DATA 201, 58, 240, 1, 96, 238	<150>	1236 DATA 6, 212, 141, 13, 212, 169	<205>
1080 DATA 139, 5, 169, 48, 141, 140	<67>	1238 DATA 17, 141, 4, 212, 169, 33	<205>
1082 DATA 5, 173, 139, 5, 201, 58	<127>	1240 DATA 141, 11, 212, 141, 20, 212	<223>
1084 DATA 240, 1, 96, 238, 139, 5	<183>	1242 DATA 174, 96, 196, 206, 97, 196	<189>
1086 DATA 169, 48, 141, 139, 5, 96	<3>	1244 DATA 208, 53, 189, 98, 196, 141	<66>
1088 DATA 173, 8, 220, 240, 53, 201	<134>	1246 DATA 7, 212, 141, 14, 212, 141	<22>
1090 DATA 2, 240, 9, 201, 5, 240	<31>	1248 DATA 0, 212, 189, 99, 196, 141	<212>
1092 DATA 45, 201, 7, 240, 1, 96	<156>	1250 DATA 8, 212, 10, 141, 15, 212	<4>
1094 DATA 169, 40, 141, 254, 7, 169	<232>	1252 DATA 10, 141, 1, 212, 189, 100	<89>
1096 DATA 42, 141, 255, 7, 169, 15	<166>	1254 DATA 196, 141, 97, 196, 232, 232	<8>
1098 DATA 141, 250, 7, 141, 251, 7	<253>	1256 DATA 232, 142, 96, 196, 173, 97	<158>
1100 DATA 169, 34, 141, 252, 7, 173	<124>	1258 DATA 196, 208, 10, 169, 0, 141	<140>
1102 DATA 242, 195, 208, 6, 169, 36	<32>	1260 DATA 96, 196, 169, 70, 141, 97	<228>
1104 DATA 141, 253, 7, 96, 169, 38	<93>	1262 DATA 196, 76, 49, 234, 0, 10	<149>
1106 DATA 141, 253, 7, 96, 169, 41	<230>	1264 DATA 30, 25, 15, 135, 33, 15	<184>
1108 DATA 141, 254, 7, 169, 43, 141	<125>	1266 DATA 62, 42, 15, 60, 50, 23	<216>
1110 DATA 255, 7, 169, 32, 141, 250	<50>	1268 DATA 0, 0, 8, 60, 50, 23	<19>
1112 DATA 7, 141, 251, 7, 169, 35	<215>	1270 DATA 0, 0, 8, 193, 44, 23	<134>
1114 DATA 141, 252, 7, 173, 242, 195	<36>	1272 DATA 0, 0, 8, 193, 44, 23	<136>
1116 DATA 208, 6, 169, 37, 141, 253	<175>	1274 DATA 0, 0, 8, 62, 42, 15	<72>
1118 DATA 7, 96, 169, 39, 141, 253	<207>	1276 DATA 162, 37, 15, 62, 42, 15	<224>
1120 DATA 7, 96, 173, 237, 195, 201	<169>	1278 DATA 193, 44, 15, 62, 42, 23	<204>
1122 DATA 10, 176, 1, 96, 238, 239	<50>	1280 DATA 0, 0, 8, 162, 37, 23	<135>
1124 DATA 195, 173, 239, 195, 201, 4	<189>	1282 DATA 0, 0, 8, 193, 44, 23	<147>
1126 DATA 240, 1, 96, 169, 0, 141	<85>	1284 DATA 0, 0, 8, 193, 44, 23	<149>
1128 DATA 239, 195, 206, 8, 208, 96	<103>	1286 DATA 0, 0, 8, 62, 42, 23	<69>
1130 DATA 173, 237, 195, 201, 15, 176	<124>	1288 DATA 0, 0, 8, 62, 42, 23	<71>
1132 DATA 1, 96, 238, 244, 195, 173	<69>	1290 DATA 0, 0, 8, 162, 37, 82	<219>
1134 DATA 244, 195, 201, 2, 240, 1	<114>	1292 DATA 0, 0, 8, 193, 44, 23	<157>
1136 DATA 96, 169, 0, 141, 244, 195	<229>	1294 DATA 0, 0, 8, 193, 44, 23	<159>
1138 DATA 238, 12, 208, 96, 173, 9	<34>	1296 DATA 0, 0, 8, 62, 42, 23	<79>
1140 DATA 220, 41, 15, 208, 5, 169	<191>	1298 DATA 0, 0, 8, 62, 42, 23	<81>
1142 DATA 1, 141, 240, 195, 173, 240	<181>	1300 DATA 0, 0, 8, 162, 37, 82	<229>
1144 DATA 195, 208, 1, 96, 238, 10	<117>	1302 DATA 0, 0, 8, 162, 37, 15	<174>
1146 DATA 208, 173, 10, 208, 240, 1	<183>	1304 DATA 135, 33, 15, 162, 37, 15	<1>
1148 DATA 96, 169, 0, 141, 240, 195	<173>	1306 DATA 62, 42, 15, 162, 37, 23	<16>
1150 DATA 169, 36, 141, 253, 7, 169	<53>	1308 DATA 0, 0, 8, 135, 33, 23	<114>
1152 DATA 1, 141, 9, 220, 169, 0	<173>	1310 DATA 0, 0, 8, 135, 33, 0	<38>
1154 DATA 141, 242, 195, 96, 162, 38	<170>	1312 DATA 120, 169, 49, 160, 234, 141	<212>
1156 DATA 142, 253, 7, 173, 21, 208	<118>	1314 DATA 20, 3, 140, 21, 3, 88	<248>
1158 DATA 41, 253, 141, 21, 208, 32	<7>	1316 DATA 169, 0, 141, 24, 212, 141	<114>
1160 DATA 115, 193, 32, 33, 194, 32	<145>	1318 DATA 96, 196, 96, 34	<160>
1162 DATA 33, 194, 169, 1, 141, 242	<108>	1320 REM ***** DATA MASCHINENPROGRAMM ***	<18>
1164 DATA 195, 96, 162, 5, 32, 4	<213>		



# COLOSSUS-MÜHLE

## Ein Strategiespiel für Ihren Commodore 64

**Nach zahlreichen Weltraumabenteuern haben wir uns diesmal auch etwas für unsere Freunde der Strategiespiele ausgedacht. COLOSSUS MUEHLE ist ein ganz besonderer Leckerbissen, denn hier können Sie Ihr strategisches Geschick beweisen.**

Ihr Gegner heißt C-64 und dürfte sich nur schwer besiegen lassen. Das Programm beherrscht die allgemeinen Regeln des Mühlespiels, da diese wohl weitestgehend bekannt sind, ersparen wir uns eine Spielerklärung. Das Programm beruht auf modernster Programmierung: Zuggenerator mit Minimax-Suche, Stellungsbewertung und ähnliches. **COLOSSUS MUEHLE** gestattet 3 Spielstufen, wobei die Suchtiefe und die Bedenkzeit in den höheren Stufen ansteigt. Die erste Stufe dürfte für den Mühle-Anfänger ausreichend sein, wobei eine durchschnittliche Bedenkzeit von 10 Sekunden auftritt. Das Programm ist vollständig in MC-Code abgedruckt, um die Bedenkzeit auf ein Minimum zu beschränken. Die Züge werden einprogrammiert, indem erst die Koordinate der zu bewegend Figur und danach die Koordinaten der Endposition eingegeben werden. Abgeschlossen wird die Eingabe mit RETURN. Wird eine Mühle geschlossen, so muß eine dritte Koordinate, die zu eliminierende Figur festlegen. In diesem Fall benötigt man keine RETURN-Taste.

Der abgedruckte MC-LOADER enthält natürlich Prüfsummen um eventuell auftretende Fehler festzustellen, trotzdem sollten Sie bei der Eingabe sorgfältig vorgehen.

### Eingabehinweise für Diskettenbenutzer:

1. MC-LOADER abtippen und abspei-

chern

2. Formatierte Diskette einlegen
3. MC-LOADER mit „RUN“ starten
4. Im Menue „D“ für Diskette wählen und Taste betätigen
5. Wartezeit ca. 15 Minuten
6. Erscheint eine Fehlermeldung, dann Zeile korrigieren und wieder mit Punkt 1 beginnen.
7. Das fertige Programm wurde auf Diskette abgelegt und kann jederzeit durch LOAD„MUEHLE“ und RUN wieder gestartet werden.

### Eingabehinweise für Kassettenbenutzer:

Kassettenbenutzer haben bei der Erstellung einer lauffähigen Mühle-Version etwas mehr Zeit aufzubringen als Diskettenbesitzer, da sich hier eine Programmdatei nicht direkt erzeugen läßt.

1. MC-Loader abtippen und abspeichern
2. Leere Kassette einlegen
3. MC-LOADER mit „RUN“ starten
4. Im Menue „K“ für Kassette wählen und Taste drücken
5. Kassette einlegen und PLAY&RECORD Taste betätigen
6. Wartezeit ca. 20 Minuten
7. Erscheint eine Fehlermeldung dann Zeile korrigieren und bei Punkt 1 fortfahren
8. Kassette zum Anfang zurück spulen und Taste betätigen
9. PLAY-Taste betätigen

10. Wartezeit 5 Minuten

11. Das fertige Programm ist im Speicher und kann nun mit SAVE „MUEHLE“ auf Kassette abgelegt werden.
12. Das spätere Wiedereinladen erfolgt einfach durch LOAD

## Versuchen Sie sich im Wettstreit mit Ihrem C-64

### Aufruf an alle Autoren!

**Wir suchen noch Strategiespiele für die Systeme Apple / C-16 / C-64 / CPC-464 / MSX / TI99 / VC-20 / ATARI in Basic und Turbo Pascal. Wer also REVERSI, SCHACH oder ähnliches im Petto hat, sollte einen Umschlag nehmen und sofort an TRONIC-VERLAG Programmierabt. (Strategiesoft), Am Stad 35, 3440 Eschwege, schicken.**

C  
o  
m  
m  
o  
d  
o  
r  
e  
64

```

3500 REM *****
3510 REM *   COLOSSUS MUEHLE V 5.0   *
3520 REM *       (C) FRANK BRALL       *
3530 REM *           1986               *
3540 REM *****
3545 REM
3546 PRINT"*** COLOSSUS MUEHLE 5.0 *** F.BRALL"
3548 PRINT"  DISKETTE <D> ODER KASSETTE <K> "
3549 GET D$:IF D$<>"K" AND D$<>"D" THEN 3549
3550 IF D$="D" THEN DT$="DISKETTE"
3551 IF D$="K" THEN DT$="KASSETTE"
3552 PRINT"LEERE "DT$" EINLEGEN UND"
3553 PRINT"TASTE BETAETIGEN !"
3554 GET E$:IF E$="" THEN 3554
3555 PRINT"BITTE WARTEN ICH ERSTELLE MC-CODE !":IF D$="K" THEN 3559
3556 OPEN 1,8,1,"@:MUEHLE,P,W"
    
```

D  
a  
s  
L  
i  
s  
t  
i  
n  
g



```

3557 PRINT#1,CHR$(2049-INT(2049/256)*256);
3558 PRINT#1,CHR$(INT(2049/256));:GOTO 3560
3559 OPEN 1,1,1,"HELP":REM KASSETTE HILFSDATEI
3560 Z=5000-1:B=0
3580 READ A$
3590 I=1:S=0:Z=Z+1:POKE 53280,Z AND 15
3600 X$=MID$(A$,I,2)
3605 IF X$="--"THEN 3630
3610 B=B+1:GOSUB 4000:I=I+2:S=(S+X) AND 255
3615 PRINT#1,CHR$(X);
3620 GOTO 3600
3630 I=I+2:X$=MID$(A$,I,2):GOSUB 4000
3640 IF X<>S THEN PRINT:PRINT"*** FEHLER IN ZEILE "Z:END
3650 IF B< 11435 THEN 3580
3660 CLOSE 1
3670 PRINT:IF D$="K" THEN 3690
3675 PRINT"DAS PROGRAMM IST UNTER DEM NAMEN"
3676 PRINT"MUEHLE ABGESPEICHERT !"
3680 END
3690 REM KASSETTE
3700 PRINT"KASSETTE BITTE ZURUECKSPULEN UND"
3705 PRINT"DANACH TASTE BETAETIGEN !":POKE 198,0
3710 GET E$:IF E$="" THEN 3710
3800 PRINT"POKE 43,0:POKE 44,144:POKE144*256-1,0:NEW"
3801 PRINT"1 OPEN 1,1,0,"CHR$(34)"HELP"CHR$(34)":FOR B=0 TO 11435:GET #1,P$
3802 PRINT"2 POKE 2049+B,ASC(P$+CHR$(0)):NEXT B:CLOSE 1"
3803 PRINT"4 ?"CHR$(34)"PROGRAMM KANN MIT SAVE ABGESPEICHERT WERDEN"CHR$(34)
3804 PRINT"3POKE43,1:POKE44,8:POKE45,154:POKE46,52"
3805 PRINT"RUN"
3809 POKE 631,19:FOR I=632 TO640:POKE I,13:NEXTI:POKE 198,8
3810 END
4000 X=0:FORU=1 TO LEN(X$):X0=ASC(MID$(X$,U,1)):X=16*X+X0-48+(X0>64)*7:NEXT U
4001 RETURN
5000 DATA 1A0B01009E32303736204652414E4BAB4252414C4C2038360000004C410B4C--AC
5001 DATA B50C660976093B1624141714C014C314320B550BEE11151273156615481548--79
5002 DATA 15B0154F0BA915FF10EE10DC10CD10CD102611CD10CD10CD10CD10CD10--DA
5003 DATA CD10CD101E112D113311CD10C1104711C211C211BA11C215F5159015BE0CAB--DD
5004 DATA 126A126A126A12D912BB12BB12EA126A12EA12E212621162115711AB0C2B12--B5
5005 DATA 2B12EE11EE11AE0CB10C1A13551355132513D013251325135513551305130E--30
5006 DATA 1308131113D916DE16DA164617FA127B177B177B177E177B177B177B177B17--21
5007 DATA 7B177B177B177B177B177B17B1172703000010EB640A6A532B127BB8B8BABB--EB
5008 DATA BFA0012CA002A9202464100520720CA92D99FE008471A200B662A564D006A5--A2
5009 DATA 65F046A202BDE808B522BDEC08B523A0FF38CBA565E5238565A564E522B564--80
5010 DATA B0F1A56565238565A56465228564A562D00598F00CE6629B0930A47199FF00--CC
5011 DATA E671EBE00490C0A471A56509304C04BFA50E100620720C4CBE0C20B4BF4CBE--F4
5012 DATA 0CA50E3003201B10A56449FF8564A56549FF85654CBE0CA000B16F18716490--06
5013 DATA 034C58B62075B4207AB6A451D005A55020DBB6A002B150B52388B150B5228B--69
5014 DATA B150208CB6A470D005A56F20DBB64CCAB4C007B0C4A465D00320B40BA56FA4--89
5015 DATA 7020AAB6866CB846DA002AA38E561F00BA0049004A001A6618466A0FFEBC8CA--78
5016 DATA D004A566100CB16CD162F0F2A901B002A904253CF002A9FFAA4C630EA56E09--B4
5017 DATA 7F256AB56AA969A000205BBC18AAEBD00138BA2A10DDC00A9058C00CB076BA--68
5018 DATA 102FA000B469846AB46B846EA29020FA0AA56CD00BA28BA56DF016846D3006--43
5019 DATA CA066D2A10FAB56AA56D856B8669846CB46DA50E1003208510A43CC00790A0--84
5020 DATA B9E9088555B9EE08B556A56E4566856FA5616C5500E40ED012BA10DEA53CC9--7D
5021 DATA 07B00620130B4C1C0A4CC50B8A10C520AE0FA50E108F30E58A300320CE0AA5--86
5022 DATA 0E3003201B10463CB00CA56C25648564A56D2565900AA56C05648564A56D05--FD

```

# Commodore 64

```

5023 DATA 65856560A5693006A900856BF01138E99030034C48B2AAA56A4A666BE8D0FA--3D
5024 DATA 066E900249FF856CA56B900249FF856D60A56C1014856EA56C49FF856CA56D--FF
5025 DATA 49FF856DE66DD002E66C60A200A56CC564D006A56DC565F00BCA56C456430--AE
5026 DATA 06B002A2018A60B0FA90FA20600BA5398550A53A8551A006D00520960BA00A--52
5027 DATA B98417BE85174C791520A50B4C9E0CA5508539A551853A4C9E0CA205BD8417--AB
5028 DATA 952DCA10F8A62EB660855FE430D004C52FF013A001A900C8915FC006D0F998--3A
5029 DATA 655F90E6EBB0E1A537A4388533843420E7FF68A868A2F89A489848A900853B--1E
5030 DATA B511AD8C178541AD8D17854260A465D00EA56420DBB6A662A463A5614CC3B6--9F
5031 DATA A56160F0104AB0453BA56DE565AAA56CE5644CE10B18A56D6565AAA56C6564--ED
5032 DATA B5648665700160A2008662B66386669005C66620720CA900850EB5704CDBBC--77
5033 DATA A5620563D0F1A56430EDA566306460A564456C8566A564100320720C20FA0A--D8
5034 DATA A564A665C56C900AA46C856CA56DB66DAA9885288629A90085628563856485--69
5035 DATA 65856A856BA5280529F0B746286629901918A565656D8565A564656C8564A5--D2
5036 DATA 63656B8563A562656A8562066D266C266B266A90D1A56449FF8564A56549FF--EB
5037 DATA B565E665D002E66460A539A63A857A867B853D863E4CE1A76868A57AA67B85--5E
5038 DATA 39863A24115018A591C97FD01220B7AB38B0042044A6182041A868682068AB--8F
5039 DATA A000843BE639D00BF00718653985399002E63AA000B139853C3047F0AEC90E--AD
5040 DATA 90100AAABD06088555BD07088556C86C5500AB68AA300ED01868856E688569--BA
5041 DATA 68856A68856B68856C68856D20260A4CBE0C68856F68857020C4094CBE0C20--A4
5042 DATA E0A94C9E0CC9E6B00A243C5008A60EB60C7027F0EAA63BF01FA50ED00320AE--9B
5043 DATA 0FA56548A56448A60ED00CA56348A56248A56148A566488A48A001843BA001--55
5044 DATA A200A53C293FC920B00AB5470A0A0A3BE547B03DC926B059C92490034C6F0E--0A
5045 DATA 2903C9039008B139E639D002E63AAAB139E639D002E63AB6488547B6600A26--E0
5046 DATA 600A26600A266038E547855FA560E548AA55F18652DB55FBA652EB560AAA5--87
5047 DATA 5F69029001E8854786488549864A204C0F4CBE0C243CC930900C290F500209--94
5048 DATA 1020630E4CBE0C7033C92AB029C9279006D00EB139AAC8B13920630EC8984C--0D
5049 DATA C40CA5396900A43A9001C820A2BB840D840EA9064CC40C20370E4CBE0CC928--25
5050 DATA 900429071008B139E639D002E63AB561840EE639D002E63AA5398562A53A85--89
5051 DATA 6320CAB4A5614CC400C860D860ED007A9ABA0AE4CA2BBC92E900AB40ED0034C--AB
5052 DATA 08AF4C48AFC92C9008D0034C7BAF4C08AFA5901001CAB6648565A980850E0A--7D
5053 DATA B50D60F009B139AE639D002E63AB66AB139E639D002E63A0A266A38652F85--66
5054 DATA 69A56A6530856A88B169856B652FB55FC8B169AA653085608BA056BF04AB15F--50
5055 DATA 854688B15F8545A004B15F850B0A6905655F8558A9008571857265608559C8--8B
5056 DATA 68301768856E68856968856A68856B686820CE0AA56D48A56C4868856CD15F--13
5057 DATA 900BD006C868D15F90054C45B2C868856DAAA57205711BF00A204CB38A656D--8F
5058 DATA AA98A422656C86718572C60BD0B38AA6720A267224453010244630030A2672--47
5059 DATA 6571A88A65728572986558A8A5726559AA984CB80DBA20B0B74C450F20F3BC--7B
5060 DATA 240E303C4CDOBBA000840D840ECB243CB15F30137007A547A4484CA2BBA50C--5C
5061 DATA 10032085104CDOB88B15F101AA980850E700AB1478564C8B147856560A50C--33
5062 DATA 3003201B104CCAA9C60DE60E50084C2CAAD0FB4CE0A9A54785648665B16485--7B
5063 DATA 61C8B1648562C8B16485636024701003201BBCA561D0048564F020102B38E9--17
5064 DATA 901026AAA5640565D01FA5638528A5624A6628B014E8D0FB8564A528856524--9E
5065 DATA 66100320720CA980850E60681869018555686900855668850ED00E85706885--5E
5066 DATA 666885616885626885636885646885656C5500A50CC50EF0D2ABF06AA00024--C3
5067 DATA 701003201BBCA5613006F08FA9F0300E38E9903009D049981046A5623014AA--F2
5068 DATA 0666089003204DB9A56228086A6663E8D0FB288564A5638565A980850E60A5--0B
5069 DATA 0E300B2466301BA0B0201D103004A56430104C04BBA50E3003201B10A665A5--F1
5070 DATA 64F0424C48B2A50E1037A000B4618462B4638466A5641005856620720CA290--EB
5071 DATA A564D008A288A565F01884653006CA06652A10FAB562A56585638661846484--D1
5072 DATA 65B470A900850E6020B00BF08BA000B1624C2B1120B110A53C1869140A20D5--D6
5073 DATA AF4CBE0CA50E300446661007A564100320720C4CBE0CA50E30F92470100320--E2
5074 DATA 1BBC20CCBC4CBE0CA50E1015A2FFA5643006EB0565F001E88AA200AB1015CA--3B
5075 DATA 3012202B8C4C1011205B10A000B1142CA5CAA2004CD20D20B00B4C2B11A50E--21
5076 DATA 100820FA08206FB4100E20441110092068B4207110205411A901850E4CBE0C--DE
5077 DATA 20EFB620711020C6FF86134CBA0C20E4FF463C90074820CCFF851368A60D30--2B
5078 DATA 1EC9309004C93A9001BA290FB5658664240E300520660FF00320CAA94C9E0C--4D
5079 DATA AAF01AA46188F009205411202CAA4C9E0CA563C53290F18A91624C9E0CA561--A4

```



# Commodore 64

```

5080 DATA FOEDB66120CAB4D0E3207110866A20EB0F20711068850E68855068855120D4--CB
5081 DATA 114CBEOCA000A53CC9359006D0088A4C2FB78A4C03B7A56A85658A4C4BB720--51
5082 DATA 5B1020EB0F207110A43CC04C90128649D003AAF00620EB0F207110203CB8D0--95
5083 DATA 032027B84C9E0C205B10A539186901857AA53A6900857B2030E14C960C88B1--82
5084 DATA 411005A20D4C37A4AAA541690185628522A542690085638523388A65418541--C2
5085 DATA 9002E6428661463CB00EA9618564A565846520940F4C9E0C203B0F4C9E0CA5--85
5086 DATA 0E301DF020A465D007A919B51620B90BA661A000EBCAF020B16220D2FFC8D0--91
5087 DATA F520FD08D00320DDBDA200BD0001F00620D2FFE8D0F5203BABA53CC93D903E--56
5088 DATA D02E3820F0FF9B38E90AB0FC49FF69011014207110463C900EBA4820F0FFB5--B1
5089 DATA 096B38E5099018AAE8CAF013203BABD0FB20D7AAA53CC943900520CCFFB513--12
5090 DATA 4C9E0C20711020C9FF8613A53CC94490EFB0DF2071108AEA20CCE14C9E0CA9--95
5091 DATA 402CA9800511D007A9BF2CA97F251185114CBEOCA539853DA53AB53E4C9E0C--58
5092 DATA 20F9ABA513F006A5902903D01DAD0002D018A513D00E241110422072AB8539--6F
5093 DATA 843A4CB10CA5902940F0D6867A847BD016A543857AA902857B207900D009A5--B9
5094 DATA 13D0BF2045ABD0BA207300240D303820420F207900F012C92CF00E2062ABA5--39
5095 DATA 3DA43E8539843A4C9E0CA47AB443A63CE055B011A613F00720CCFFB513F006--77
5096 DATA AAF00320F4AC4C9E0C8507C922F007A93AB507A92C188508A56548A57AA47B--3A
5097 DATA 6900208DB4A671867A6820940F4C7513A53C6988853DA2058655A0AFAD0102--F5
5098 DATA 913CA940A0AC913CAE0202A901200F140AD00F900D2A200F143007CAF00BC9--26
5099 DATA 00F0E9C655D0D54CAE0CC565D0F54CBE0C913C4A913CB13C60A50E850C20EB--24
5100 DATA 0F2012104C3A14A50E850C3009A9BCA0B920A2BB3007A2018665CAB664200F--4B
5101 DATA BC20EB0F201210209714D0068A18650BAA9ABAE050B0034C35A4A50CF00FA5--26
5102 DATA 6548A56448A56D48A56C484C8514A566097F25628562A97CA014852284234C--0A
5103 DATA 43AE20FCBB202BBC203BAEA53A48A53948A54A48A54948A50C484C9E0CBAE8--20
5104 DATA EBBD01010AD01FA009B002A010840BA54AF013DD0301D007A549DD0201F007--A9
5105 DATA 8A18650BAAD0DB6088844A209714F0034C30AD9ABD0301854ABD02018549A0--5C
5106 DATA 016848850E30278A18690648690685246820A2BBBABD0B018566850CA549A4--82
5107 DATA 4A2067B820D0BBA001205DBC4C2D15BD070118714991498565BD06011002A0--6D
5108 DATA FF840CA000714991498564BD0901856DBD0801856C20130R38BAE50CF00DBD--F7
5109 DATA 0501853ABD040185394C9E0C8A18650BAA9AD0F5C63BB139856B207110BAF0--4C
5110 DATA 050AC56B9004A56BD062A8463CB011A56BD002A903653948A53A690048A9BD--3A
5111 DATA 48B139A8CB1398539863A4CCD0CA9FFB54A2097149A68C98DF0134CE0AB68--0A
5112 DATA 854968854AA000689149C8C005D0F868853A6885394CCD0CC63BA50E300BA5--3F
5113 DATA 61F00AA902D008A5640565D0F6B1394CC40CB139652D8564C8B139652E8565--B1
5114 DATA CBB139652D856CC8B139652E856DA9056539B56AA53A6900856BB96A009164--F7
5115 DATA 8B10F8A005D0C8CB139652D854788B139652E8548A539690348A53A690048--D1
5116 DATA B147D0034CAEB3853A88B1478539A002B1478549C8B147854AC8B149488810--0E
5117 DATA FAA54A48A54948A50E20640F4C9E0CB139840C850BC8B1398545C8B1398546--A6
5118 DATA A531855F38E52F8569A5328560E530AA0569F021A00518B139652F856B88B1--52
5119 DATA 396530856CA001B16B88116BD05EA569916BC88A916B2094B12008A4A00084--5B
5120 DATA 3B8472A205A545915F1001CAC8A546915F1002CACAB671A50BA004915FD003--E4
5121 DATA 20EB0F8422205B10A422A665C8EBD003186901915FC88A915F204CB3A42286--BE
5122 DATA 718572C60BD0DA20AAB2A9064CC40C4C4DB2B8840AA001201517460CAAF015--79
5123 DATA 205AE2C60BF00E202E1786BAC60BF005202E1786B92001174C9E0CA53CC95E--70
5124 DATA 9005D0094C59E1206FE14CAEA74C6FE1A20086B786B98690840CB4BA847BB1--53
5125 DATA 39850BE639D002E63A606818690185456869008546460CB00320EB0F207110--4A
5126 DATA 6C4500201517202E1786B8C60BF020202E1786BAE0039002C6B9C60BF01120--0C
5127 DATA 2E1786B9C60BF00820EB0FC60D205AE220C1E14C9E0C4C0BAF4C0BAF4C0BAF--57
5128 DATA 4C0BAF3233AC34AC3490179817941AF21001DAAA0000160132013202313001--93
5129 DATA 30013001330131013301350130013201320231350130013001320135023131--5F
5130 DATA 01300130013401320134013601380132013502313401300130013201380231--40
5131 DATA 32013001300133013501370139013001320138023133013001300133013102--3D
5132 DATA 31310232320130013401340231300231320231390133013702313102313601--D1
5133 DATA 30013301390231340231380130013401360231330231350232310133013302--AA
5134 DATA 31340232340130013202313202313701300130013302313602313802323001--CF
5135 DATA 30013202313302313701300130013202313102323001300130013402313702--93
5136 DATA 31390232310232330132023134023230013001300132023130023233013001--C5

```

[illegible]



# Commodore 64

```

5194 DATA 4547494E4E454E202B4A2F4E293C943CEA203F3E3F3F3F3F3F924792EB02--9E
5195 DATA 1F05191F9992E94A0592E94E050C1F05191F6592E94A021F04B1CB92E94E02--8B
5196 DATA 1F04B0C88B8B80000000000721C88B80041F05191F6588B1011F05191F65E9--1C
5197 DATA 933CB1C5B8851185B1A408B185B2A408E40985B1A408B285B3A408E40985B2--92
5198 DATA A408B185B1A408E40985B2A408B285B3A408E40985B3A408B185B1A408E409--F6
5199 DATA 85B3A408B285B2A408E4098514B9C5F0851185B1A408B185B2A408E40A85B1--9D
5200 DATA A408B285B3A408E40A85B2A408B185B1A408E40A85B2A408B285B3A408E40A--FB
5201 DATA 85B3A408B185B1A408E40A85B3A408B285B2A408E40A85141925608B0ED58B--FC
5202 DATA 20B2071B0820832082209249B28E2008D68DB3061F04B3D619209EB2912008--AE
5203 DATA D690B3061F04B3D6B0C9B0CAB1C5F885118B85A40309B2071B0820B620C920--3D
5204 DATA C189B107C98985E4041920C98AB107CAB8A5E4058514B1C5F0851185B1A408--3E
5205 DATA A40395051F0519214485B2A408A40395051F0519214485B3A408A40395051F--ED
5206 DATA 05192144B1D7B1D89BA4048597A4080820B2071B08210E211D213898B107D8--8E
5207 DATA 9889061F051920FB19214489B108C98998041F0519214498D98999119999B1--56
5208 DATA 07A404E404991497B107D797B3061F051920FB85149BB108A411DA961B0821--68
5209 DATA 5521E12287B1C5F8851185A403B0051F051921D289B0021F051921BF8B85B1--2E
5210 DATA A409A403051F051921868B85B2A409A403021F051921A08B85B1A40AA40305--6E
5211 DATA 1F051921BF8B85B2A40AA403051F051921BFB1C8898B119AB107DAB19AB0E4--FC
5212 DATA 0EB29A85E40EB39A8BA404E40EB8141921D29AB107DAB19AB0E40EB29A85E4--B6
5213 DATA 0EB39AB0E40EB5149B9AE4119A9BB108A41108DA1DB1C58A851185A405C787--74
5214 DATA B0E403B1C887A4068B11878BA407DC9CA403B0051F0519227289B0021F0519--B1
5215 DATA 225F8B9CB1A409A403051F051922268B9CB2A409A403021F051922408B9CB1--DD
5216 DATA A40AA403051F0519225F8B9CB2A40AA403051F0519225FB1DD899D119AB107--20
5217 DATA DAB19AB7E40EB29A9CE40EB39A9DA404E40E9D141922729AB107DAB19AB7E4--3C
5218 DATA 0EB29A9CE40EB39AB0E40EB814878BE40385149B9AE4119A9BB108A41108DA--B4
5219 DATA 1DB1C58A851185A405C7B1C8F8881188A403B0051F0519231989B0021F0519--87
5220 DATA 230687B0E4038B88B1A409A403051F051922C58B88B2A409A403021F051922--02
5221 DATA DF8B88B1A40AA403051F051923028B88B2A40AA403051F05192302B1DD899D--94
5222 DATA 119AB107DAB19AB7E40EB29A8BE40EB39A9DA404E40E9D14878BE403192319--4A
5223 DATA 878BE4039AB107DAB19AB7E40EB29A8BE40EB39AB0E40EB81485149B9AE411--FD
5224 DATA 9A9BB108A41108DA1D9BB108A411B107DE9BA411DF9E9F031F031DA020B102--62
5225 DATA 1F051923F49B93031F051923F49EC79F87118BB2071C0824692FBB24E51A25--9C
5226 DATA B3B487A0210EB09E40EB8B2071C0824A72FBB252287149EC59FB1088511B4--47
5227 DATA 85A40EE02185C785B107C89F8811B48BA40EA021061F051923A0B48BA40EE0--44
5228 DATA 2188C78B148785021F051923EEB185A40EE021B185B187A40EE40EB187A021--5C
5229 DATA E40EB285A40EE021B285B287A40EE40EB287A021E40EB385A40EE021B385B3--B2
5230 DATA 87A40EE40EB387A021E40EB487B485A40EE40EB514B0E0201D9EE0229EC7B2--4D
5231 DATA 9BA40FB287A40E051F0519245CB19BA40FB187A40E051F0519245CB187B1A0--2F
5232 DATA 22A40EE40EB287B2A022A40EE40EB387A40EE021B387B3A022A40EE40EB1A0--A8
5233 DATA 22B19BA40FE40EB2A022B29BA40FE40EB3A022A021E40EA022B107E022A021--DE
5234 DATA B0021F031DB7B107C7879F061F051923F91DB187A40EB0021F0519247EB187--4C
5235 DATA A40EB0E4031924868EB108CE8CB107CCB287A40EB10EE403B387A40EB0021F--7F
5236 DATA 031DB387A40EB0E4038FB108CF90B108D01DB187A40EB0021F051924BDB187--E2
5237 DATA A40EB10EE4031924C58EB107CE8CB108CCB287A40EB0E403B387A40EB0021F--BE
5238 DATA 031DB387A40EB1E4038FB107CF90B107D01DB187A40EB0021F051924FAB187--1E
5239 DATA A40EB0E40319250291B108D18FB107CFB287A40EB1E403B387A40EB0021F03--FD
5240 DATA 1DB387A40EB0E4038CB108CC8DB108CD1DB187A40EB0021F05192537B187A4--F2
5241 DATA 0EB1E40319253F91B107D18FB108CFB287A40EB0E403B387A40EB0021F031D--B3
5242 DATA B387A40EB10EE4038CB107CC8DB107CD1DB1C5B9851185A4134BF4850885A4--C7
5243 DATA 130EE4138514BAB0E413B1C5F0851185A4144BA622850885A4140EE4148514--C7
5244 DATA F1B0E414B1C5F0851185A4154BA622850885A4150EE4158514F1B0E41588B1--24
5245 DATA 071B0629922B8CB0E021B0D6B0E023B0E024B1C5F08511B0D8B0E025B1C8B3--9D
5246 DATA 8B11858BA408A403B2071B0825DD260125EFA025B0011F05192609A025B108--72
5247 DATA E02519260EA025B0041F05192609A025B107E02519260E858BA408D819260E--E4
5248 DATA B3C8B0E0258B14A02522B1071B0A26F826F8261F26DCA02520E025B1C89BA4--B2
5249 DATA 068B11988BA407E026A026A403A025051F0519268C85B1A408A026021F0519--7D
5250 DATA 268C85B2A408A026021F0519268C85B3A408A026021F0519268CB6C8A025B2--DC
5251 DATA 071B08266F268C267E96B107D68BB1021F05B1E02419268C96B108D68BB10E--8A

```

[illegible]

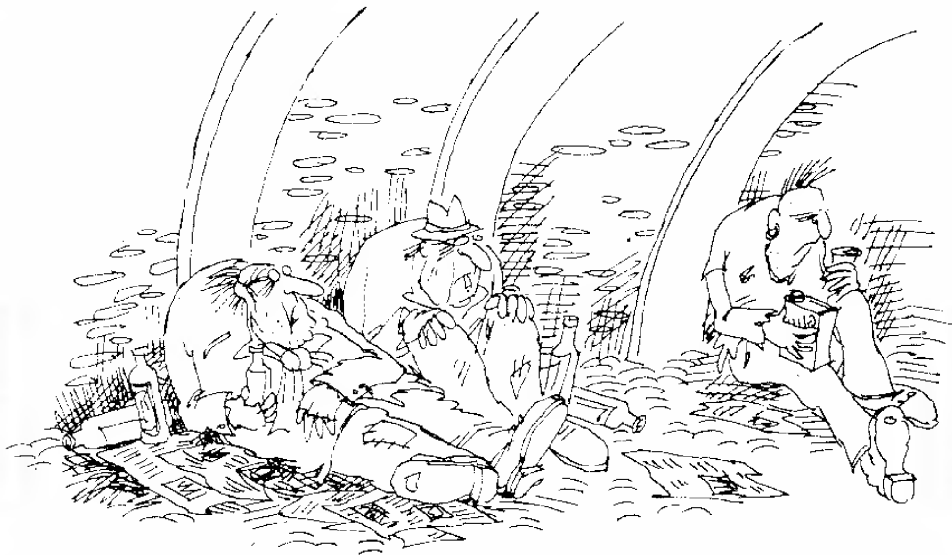




## Computronic

# JOSEF in den Katakomben

per Joystick sicher und unerschrocken durch das Höhlensystem



Auch von diesen finsternen Gestalten wird sich Josef auf seinem gefährlichen Weg durch die Katakomben nicht aufhalten lassen! - Oder? -

## Steuern Sie unseren Helden

Das ist sicherlich einer der Vorteile der Computer, daß man sich bequem in vergangene Zeiten zurückversetzen kann. So hat man beispielsweise hier die Möglichkeit, mit „Josef“ im alten Ägypten herumzuforschen. „Josef“ befindet sich in einem unterirdischen Höhlensystem, das er erforschen muß.

Der Spieler hat die Aufgabe, Josef so zu steuern, daß er alle Schätze sammelt, ohne dabei gegen einen Erdhaufen zu stoßen oder von der Leiter zu fallen, denn man besitzt nur 3 Leben. An einer bestimmten Stelle befindet sich eine Schlucht, die man erst überqueren kann, nachdem ein Symbol, das sich in der obersten Reihe befindet, erbeutet wurde. Dann nämlich schließt sich der Hohlraum von selbst und der Weg ist frei. Wenn Josef alle Schätze eingesammelt hat, erscheint ein Schlüssel, den man benutzen muß, um durch die Tür, die nun erscheint, in den nächsten Level vorzudringen.

Die Zeit, die Josef besitzt, um all seine Aufgaben zu erledigen, wird natürlich von Level zu Level kürzer.

## Ein tolles Labyrinthspiel für den VC-20-Computer

```
0 PRINT" (CLEAR) ":K=7176:POKE36879,8:W=4 <7>
1 PRINT" (DOWN2 RIGHT) RVSON GREEN>ELCHSD
FT (DOWN LEFT) PRESENTS: ":FORT=1TO3000:NE
XT:POKE36869,255 <52>
2 FORT=1TO18 <241>
3 FORT=1TO7:READA:POKEK+I,A:NEXTI <115>
4 K=K+8:NEXTY <6>
5 FORT=768TO8185:POKET,8:NEXTT <150>
10 PRINT" (HOME DOWN2 YELLOW) HHHHHHHHHH
HHHHHHH" <193>
11 PRINT" HHHHHHHHHHHHHHHHHH" <142>
12 PRINT" HHHHHHHHHHHHHHHHHH" <248>
13 PRINT" HHHHHHHHHHHHHHHHHH" <150>
14 PRINT" HHHHHHHHHHHHHHHHHH" <225>
15 PRINT" (DOWN2 RVSON YELLOW RIGHT) IN (D
OWN2 LEFT) DEN (SPACE) KATAKOMBNEN" <133>
16 FORT=8031TO8010STEP-1:POKET,W:POKET+3
0720,1:W=W+1:IFW>6THENW=4 <125>
17 POKEK+1,8:FORT=1TO350:NEXTI:NEXTT <55>
18 PRINT" (RVSON DOWN3 YELLOW) B (BLUE) Y (RI
GHT GREEN) M (PURPLE) I (CYAN) C (RED) H (WHITE)
A (YELLOW) E (BLUE) L (RIGHT GREEN) A (PURPLE) L
(CYAN) T" <236>
19 GETA$:IFA$="" THEN19 <55>
20 POKE36869,242:PRINT" (CLEAR DOWN7 RIGH
T) * (L) / (SPACE) ? (DOWN3 LEFT) (V) " <63>
21 GETA$:IFA$="N" THEN39 <146>
22 IFA$="Y" THENPRINT" (CLEAR) ":GOTO24 <121>
23 GOTO21 <151>
24 PRINT" * IE (SPACE) HABEN (SPACE) DIE (SPACE)
* UFGABE (SPACE DOWN) JOSEF (SPACE) SO (SPACE)
ZU (SPACE) STEuern, (SPACE2 DOWN) DASS (SPACE
```

## Listig / Teil 1

```
E) ER (SPACE) ALLE (SPACE) * CHAETZE" <17>
25 PRINT" (DOWN) SAMMELT, (SPACE) OHNE (SPACE)
DABEI (SPACE3 DOWN) GEGEN (SPACE) EINEN (SPA
CE) "RDHAUFEN (SPACE DOWN) ZU (SPACE) STOSSEN
(SPACE) ODER (SPACE) VON (SPACE) " <117>
26 PRINT" (DOWN) DER (SPACE) LEITER (SPACE) ZU
(SPACE) FALLEN, (SPACE DOWN) DENN (SPACE) MAN
(SPACE) BESITZT (SPACE) NUR (SPACE2 DOWN) 3 (S
PACE) LEBEN. " <12>
27 GETA$:IFA$="" THEN27 <49>
28 PRINT" (CLEAR) * N (SPACE) EINER (SPACE) * TE
LLE (SPACE) BEFIN- (DOWN) DET (SPACE) SICH (SPA
CE) EINE (SPACE) "LUFFT. (SPACE DOWN) M (SPAC
E) UEBER (SPACE) DIESE (SPACE) ZU (SPACE) KOM- " <105>
29 PRINT" MEN, (SPACE) MUSS (SPACE) MAN (SPACE)
ALS (SPACE) ERS- (DOWN) TES (SPACE) DAS (SPACE)
SICH (SPACE) IN (SPACE) DER (SPACE3 DOWN) OBE
RSTEN (SPACE) "EINE (SPACE) BEFIN- " <63>
30 PRINT" LICHE (SPACE) * SYMBOL (SPACE) ERGATT
ERN (DOWN) IAT (SPACE) MAN (SPACE) DIESES (SPAC
E) BETAN, (SPACE DOWN) SO (SPACE) SCHLIESST (S
PACE) SICH (SPACE) DER " <235>
31 PRINT" (DOWN) HOHLRAUM (SPACE) VON (SPACE)
SELBST, (SPACE2 DOWN) UND (SPACE) DER (SPACE)
DEB (SPACE) IST (SPACE) FREI. " <75>
32 GETA$:IFA$="" THEN32 <1>
33 PRINT" (CLEAR) IAT (SPACE) JOSEF (SPACE) AL
LE (SPACE) * CHAET- (DOWN) ZE (SPACE) GESAMMELT
, (SPACE) SO (SPACE) ER- (SPACE2 DOWN) SCHEINT
(SPACE) OBEN (SPACE) EIN (SPACE) " <20>
34 PRINT" (DOWN) * CHLUESSEL, (SPACE) DEN (SPA
CE) MAN (SPACE3 DOWN) AUCH (SPACE) NEHMEN (SPA
```





```

CE} MUSS, {SPACE} DENN {DOWN} NUN {SPACE} ERSCH
EINT {SPACE} UNTEN" <216>
35 PRINT "{DOWN} LINKS {SPACE} EINE {SPACE} | U
ER, {SPACE} DURCH {DOWN} DIE {SPACE} MAN {SPACE
} IN {SPACE} DEN {SPACE} NAECHST- {DOWN} TEN {SPA
CE} LEVEL {SPACE} EINDRINGEN {SPACE2 DOWN} KA
NN." <199>
36 GETA$: IFA$="" THEN 36 <57>
37 PRINT "{CLEAR} - INE {SPACE} WEITERE {SPACE
} SCHWIER- {SPACE DOWN} IGKEIT {SPACE} IN {SPA
CE} DIESEM {SPACE} SPIEL {DOWN} IST {SPACE} DIE
{SPACE} MEIT, {SPACE} DIE {SPACE} SICH" <43>
38 PRINT "VON {SPACE} LEVEL {SPACE} ZU {SPACE}
LEVEL {SPACE4 DOWN} VERRINGERT. {SPACE11 DO
WN} ♥ TEUERUNG: 40YSTICK" <55>
39 PRINT "{DOWN3} ": POKE631,131: POKE198,1 <165>
40 DATA24,16,56,52,48,24,48,16 <238>
41 DATA24,16,52,88,16,56,72,64 <68>
42 DATA24,16,56,84,16,24,104,8 <106>
43 DATA24,8,28,44,12,24,12,8 <236>
44 DATA24,8,44,26,8,28,18,2 <43>
45 DATA24,8,28,42,8,24,22,16 <21>
46 DATA16,56,16,56,124,56,40,108 <91>
47 DATA0,0,0,0,0,0,0,0 <6>
48 DATA255,255,255,255,255,255,255,255 <31>
49 DATA0,0,0,0,0,56,40,56 <174>
50 DATA251,251,0,223,223,223,0,247 <19>
51 DATA66,126,66,66,66,126,66,66 <213>
52 DATA0,255,255,60,60,0,255,0 <158>
53 DATA0,0,0,0,0,24,60,124 <54>
54 DATA0,0,2,116,120,255,254,252 <37>
55 DATA24,36,24,8,24,24,8,24 <47>
56 DATA255,195,165,153,153,165,195,255 <41>
57 DATA0,0,64,46,30,255,127,63 <175>
99 END <227>
100 ***** <48>
101 *****JOSEF***** <18>
102 *****IN**DEN***** <105>
103 ***KATAKOMBEN*** <112>
104 *(C)*BY*ELCHSOFT* <125>
105 WRITTER: M. ALT*** <178>
106 *FELDWEB*12***** <170>
107 *5509*HEIDENBURG* <167>
108 ***** <56>

```

## Josef in den Katakomben Zweiter Teil

```

0 SC=0:LE=3 <215>
1 POKE36879,8:PRINT "{CLEAR}":PRINT "{CYAN
}PRESS {SPACE} ANY {SPACE} KEY" <99>
2 GETA$:IFA$="" THEN2 <77>
4 ZT=500:OD=0:WW=1:BB=0:NN=15 <219>
5 POKE37139,0:A=8161:C=30720:W=4:X=7750:
M=7903:SS=36878:POKESS,15:DD=36874:OD=15
:FF=0 <149>
9 POKE36869,255 <159>
10 PRINT "{CLEAR}" <40>
11 FORT=7680T08185:POKET,8:NEXT <12>
17 REM ***** <190>
18 REM ***BILDSCHIRMAUFBAU*** <112>
19 REM ***** <192>
20 PRINT "{HOME YELLOW RVSON} SC: "SC:PRINT
 "{HOME RVSON}" TAB(OD) ZT <224>
21 PRINT "KKKKKKKKKKKKKKKKKKKKKKKKKK"; <69>
22 PRINT "KHHHHHHHHHHHHHHHHHHHHHHHKK"; <83>
23 PRINT "KHHHHHHHHHHHHHHHHHHHHHHHKK"; <78>
24 PRINT "KKKKKKKKKKKKKKKKKKKKKKKKKK"; <99>
25 PRINT "KHHHHHHHHHHHHHHHHHHHHHHHKK"; <139>
26 PRINT "KHHHHHHHHHHHHHHHHHHHHHHHKK"; <52>
27 PRINT "KKKKKKKKKKKKKKKKKKKKKKKKKK"; <86>
28 PRINT "KHHHHHHHHHHHHHHHHHHHHHHHKK"; <78>
29 PRINT "KHHHHHHHHHHHHHHHHHHHHHHHKK"; <199>
30 PRINT "KKKKKKKKKKKKKKKKKKKKKKKKKK"; <60>

```

```

31 PRINT"KHHHHHHHHHHHHHHHHHHHLHHK"; <121>
32 PRINT"KHHHHHJHHHHNNHHHJHHHHHLHHK"; <242>
33 PRINT"KKKKKKKKKKKKKKKKKKKKKKKKK"; <95>
34 PRINT"KHHHHHHHHHHHHHHHHHHHHHHHHK"; <96>
35 PRINT"KHHHJHHNNHHHLHHHHHHJHHHHK"; <211>
36 PRINT"KKKKKKKKKKKKKKKKKKKKKKKKKK"; <107>
37 PRINT"KHHLHHHHHHHHHHHLHHHHHHHK"; <135>
38 PRINT"KJHLHHHHHHHHHHHHHHHHHHHHK"; <8>
39 PRINT"KKKKKKKKKKKKKKKKKKKKKKKKKK"; <97>
40 PRINT"KHHHHHHHLHHHHHHHHHHHHHHHHK"; <86>
41 PRINT"KHHHHJHLHHHHHHHHHHHHHHHDK"; <15>
42 PRINT"KKKKKKKKKKKKKKKKKKKKKKKKKK"; <245>
43 POKE8185,11 <149>
44 IFFF=1THENPOKEM,11:POKEM+1,11:POKEM+1 <97>
+C,1:POKEM+C,1:M=8113:POKEM,8:POKEM+1,8
45 IFFF=2THENPOKEM,11:POKEM+1,11:POKEM+1 <76>
+C,1:POKEM+C,1:M=7972:POKEM,8:POKEM+1,8
46 IFFF=2THENPOKE7903,11:POKE7904,11:POKE <7>
E7903+C,1:POKE7904+C,1
192 REM ***** <117>
193 REM ***STEUERUNG*** <217>
194 REM ***** <119>
195 POKE X,8:X=X+WW:GOSUB290:POKE X,NN:IFF <80>
EEK(X+22)=12THENWW=WW*-1:POKE X,8
197 ZT=ZT-1:GOSUB370 <162>
200 J9=PEEK(37137):POKE37154,127:J8=PEEK <237>
(37152):POKE37154,255
201 IF (J9AND16)=0THENZ=-1:E=-23:R=21:W=W <208>
+1:GOSUB260:GOSUB210
202 IF (J8AND128)=0THENZ=1:E=-21:R=23:W=W <109>
-1:GOSUB263:GOSUB210
203 IF (J9AND4)=0ANDPEEK(A-22)=12THENY=-2 <70>
2:GOSUB230
204 IF (J9AND8)=0ANDPEEK(A+22)=12THENY=22 <137>
:GOSUB230
205 IF (J9AND4)=0ANDPEEK(A-1)=11ANDPEEK(A <60>
+1)=11THENY=-22:GOSUB230
207 IF (J9AND32)=0THENGOSUB250 <218>
209 GOTO195 <59>
210 POKEA,8:A=A+Z:GOSUB270:POKEA,W:POKEA <203>
+C,1
211 RETURN <97>
230 POKEA,12:A=A+Y:POKEA,7:IFY=22ANDPEEK <93>
(A-1)=11ANDPEEK(A+1)=11THENPOKEA-22,8
231 RETURN <117>
250 POKEA,8:A=A+E:GOSUB280:POKEA,W:FORT= <82>
1TO111:NEXT:POKEA,8:A=A+R:GOSUB285:POKEA <137>
,W
251 RETURN <244>
260 IFW>6THENW=4 <243>
261 IFW<4THENW=4 <149>
262 RETURN <225>
263 IFW<1THENW=3 <228>
264 IFW>3THENW=3 <152>
265 RETURN <144>
270 RR=PEEK(A) <14>
271 IFRR=11THENA=A-Z <252>
272 IFRR=14ORRR=15ORPEEK(A+22)=8THEN450 <200>
273 IFRR=10THENGOSUB300
274 IFPEEK(A-Z-22)=12THENPOKEA-Z,12:POKE <138>
A-Z+C,1
275 IFRR=17THEN350 <38>
276 RETURN <163>
280 IFPEEK(A)=13THENGOSUB295 <112>
281 IFPEEK(A)=16THENPOKE8142,17 <201>
282 IFPEEK(A+1)=11ORPEEK(A+2)=11ORPEEK(A <236>
-1)=11ORPEEK(A-2)=11THENA=A-Z*2
283 RETURN <170>
285 IFPEEK(A)=14ORPEEK(A)=15ORPEEK(A)=18 <63>
THEN450
286 IFPEEK(A)=10THENGOSUB300 <188>
287 IFPEEK(A-R+22)=12THENPOKEA-R,12 <88>
288 RETURN <175>
290 IFX=ATHEN450 <230>
292 IFWW=1THENNN=15 <15>
293 IFWW=-1THENNN=18 <23>

```



VC-20



```

294 RETURN <181>
295 POKEM,9:POKEM+1,9:POKEM+C,1:POKEM+1+ <211>
C,1:RETURN
300 POKEDD,230:FORT=1T0150:NEXT:POKEDD,0
:SC=SC+100:BB=BB+1:PRINT"<HOME RVSON>SC: <214>
"SC
301 IFBB=110RBB=220RBB=330RBB=440RBB=55T
HENPOKE7737,16 <31>
302 RETURN <189>
350 Q=100:FORT=1T0100:POKEDD,Q:Q=Q+1:NEX
TT:POKEDD,0:POKE36869,240 <236>
351 PRINT"<CLEAR DOWN7 RIGHT5>BONUS:900(<
SPACE>PKT." :FORT=1T03000:NEXT:POKE36869, <132>
255
352 POKEA,11:POKEA+C,1:A=0161:SC=SC+900:
FF=FF+1:GOSUB360:QQ=QQ+30:ZT=500-QQ:GOTO <90>
10
360 IFFF=3THENFF=0 <63>
361 RETURN <248>
370 PRINT"<HOME RVSON>"TAB(00)ZT <127>
371 IFZT<10THENQQ=17 <3>

```

```

372 IFZT<100THENQQ=16 <146>
373 IFZT=0THEN500 <145>
374 RETURN <5>
450 POKEDD,150:FORT=1T0500:NEXT:POKEDD,0 <118>
451 LE=LE-1:IFLE=0THEN500 <105>
452 GOTO4 <182>
497 REM ***** <167>
498 REM ***SPIELEND*** <109>
499 REM ***** <169>
500 POKEDD,150:FORT=1T0500:NEXT:POKEDD,0
:POKESS,0 <25>
502 PRINT"<CLEAR>":POKE36869,240:PRINT"<
RED DOWN4 RIGHT5>GAME<SPACE>OVER" <230>
503 PRINT"<YELLOW DOWN3>SCORE:"SC <230>
504 PRINT"<GREEN DOWN3>WOLLEN<SPACE>SIE<
SPACE>NOCH<SPACE>MAL<SPACE>?<DOWN2 CYAN
LEFT4>(J/N)" <176>
505 GETA#:IFA#="J"THENRUN <156>
506 IFA#="N"THENEND <96>
507 GOTO505 <83>
ENDE DES LISTINGS

```

C-16 -- ● -- C-16 -- ● -- C-16 -- ● -- C-16

C-16/116-Fans aufgepaßt! Jetzt kommt „FROGGER“ auch zu Euch!

# FROGGER

Sicher werden die meisten die Spielidee von Frogger schon kennen. Ein Frosch muß über eine stark befahrene Straße und einen verseuchten Fluß (Umweltschutz läßt grüßen!) gesteuert werden.

Als erstes muß man vom Bürgersteig auf die Straße springen und sich unbeschadet

an den Autos vorbeischlängeln, um so die andere Seite zu erreichen. Nun gilt es den Fluß zu überqueren. Dabei sind einem schwimmende Baumstämme und Schildkröten behilflich, über die man hinweghüpfen kann. Natürlich darf man nicht in das Wasser plumpsen!

Ist alles überstanden, darf man ins ersehnte Heim springen und sich eine Fliege gönnen. Hat man alle 7 Häuschen mit Fröschen besetzt, ist die erste Schwierigkeitsstufe geschafft. Je schneller man das Ziel

erreicht hat, desto mehr Punkte werden dem Spieler gutgeschrieben. In den höheren Levels hat man dann immer weniger Zeit, um sein Ziel zu erreichen. Insgesamt besitzt man 4 Froschleben. Sind alle Leben verbraucht und mehr als 10.000 Punkte erreicht, darf man sich in die Highscore-Liste eintragen. Anschließend kann das Spiel mit der Space-Taste neu gestartet werden.

Die Joysticksteuerung erfolgt an Port I. Mit der Feuertaste kann der Frosch zum Springen gebracht werden.



Keine Angst! Ihr Widerpart ist nicht ganz so gewaltig, wie auf dieser Abbildung hier!





## Start des Listings: Frogger

```

10 POKE54,47:V=65280:PRINT"(CLEAR)":COLOR4,1 <154>
20 POKEV+18,PEEK(V+18)AND251 <82>
30 POKEV+19,PEEK(V+19)AND30R48 <79>
40 RESTORE540:FORT=832T0849:READA:POKET, <138>
A:NEXT:SYS832
50 RESTORE550:FORT=12800T013320:READA:IF <25>
A>-1THENPOKET,A:NEXT
60 PRINT"(CLEAR DOWN3 BLACK)DELETE540-10 <99>
20" <73>
70 PRINT"(DOWN2)DELETE-80"
80 PRINT"(DOWN2)RUN":CHAR1,10,20,"WHITE <250>
">DRUECKE(SPACE)3X(SPACE)>RETURN<(BLACK)" <30>
:PRINTCHR$(19):END
90 B=3930:A=0:HC=0:M=0:VOLB:X=917
100 MA=100000:MB=75000:MC=50000:MD=25000 <124>
:ME=15000:MF=10000
110 MA$="X" L:MB$="I" O:MC$="I" N: <206>
MD$="MICKEY":ME$="I" MF$="LUTZ" <214>
120 PRINT"(HOME2 CLEAR)":SOUND1,500,5
130 PRINT"(RIGHT9 SPACE3 D.BLUE)"(SPACE1 <122>
4 PURPLE)"(SPACE) (SPACE2)"
140 PRINT"(RIGHT9 D.BLUE)"(RED)"(GREEN) <181>
"(ORNG)"(CYAN)"(CYL.GRN)"(PURPLE SPACE2)
"(SPACE)"
150 PRINT"(RIGHT9 L.BLU)"(PINK)"(L.GRN) <50>
"(YELLOW)"(WHITE)"(PURPLE SPACE3)"(SP <248>
ACE)"(SPACE)"
160 FORI=1T0150:NEXT
170 PRINT"(DOWN2 RIGHT12 L.BLU)"L:GOSUB <233>
200:PRINT"(RIGHT)"I:GOSUB200:PRINT"(RIG <64>
HT)"I:GOSUB200
180 PRINT"(RIGHT)"I:GOSUB200:PRINT"(RIG <224>
HT)"I:GOSUB200:PRINT"(RIGHT)"I:GOSUB20 <248>
0:PRINT"(RIGHT)"I:GOSUB200:PRINT"(RIG <207>
T0210 <76>
200 SOUND3,1010,3:FORI=1T070:NEXT:RETURN <112>
210 COLOR1,6,3:CHAR1,10,6,"-":CHAR1,28,6 <142>
,"-": <178>
220 COLOR1,6,0:CHAR1,5,9,"(SPACE35)" <209>
230 COLOR1,6,3:CHAR1,5,10,"(SPACE35)"
240 COLOR1,6,4:CHAR1,5,11,"(SPACE35)"
250 COLOR1,6,6:CHAR1,5,12,"(SPACE35)"
260 COLOR1,6,7:CHAR1,5,13,"(SPACE35)"
270 RESTORE290:FORI=1T072:READW:POKEW,67 <209>
:X=X-5:SOUND2,X,2:SOUND2,X-107,1:POKEW,1 <45>
08
280 NEXTI
290 DATA3437,3438,3439,3477,3517,3518,35 <176>
57,3597,3441,3442,3443,3481,3483,3521
300 DATA3522,3561,3563,3601,3603,3446,34 <54>
47,3485,3488,3525,3528,3565,3568,3606
310 DATA3607,3451,3452,3453,3490,3530,35 <164>
32,3533,3570,3573,3611,3612,3456,3457
320 DATA3458,3495,3535,3537,3538,3575,35 <182>
78,3616,3617,3460,3461,3462,3500,3540
330 DATA3541,3580,3620,3621,3622,3464,34 <123>
65,3466,3504,3506,3544,3545,3584,3586
340 DATA3624,3626 <167>
350 FORI=1T0150:NEXTI <218>
360 CHAR1,5,16,"(CYL.GRN)"(RED)1985(SPACE <135>
D.BLUE)"(SPACE)"(SPACE)"(SPACE)"(SPACE)
I"(SPACE)"I"
370 PRINT"(DOWN RIGHT15 CYAN)FUER(SPACE) <65>
COMMODORE(SPACE)"-16(DOWN2 LEFT17 PINK) <242>
I:05977/1753"
380 READA,B,C:C=C*5.1
390 IFA=-1THENGOTO500 <44>
400 SOUNDA,B,C:GOTO380 <208>
410 DATA1,897,2.5,2,516,1,1,897,2.5,2,51 <120>
6,1,1,864,2,2,383,1,1,897,2,2,516,1
420 DATA1,897,2,2,516,1,1,881,2,2,453,1,

```

## Start des Listings: Frogger

```

1,864,2,2,383,1,1,864,1,2,383,1 <225>
430 DATA1,864,1,2,383,1,1,864,1,2,383,1, <35>
1,897,2,2,516,1,1,881,2,2,453,1
440 DATA1,854,2,2,345,1,1,810,2,2,169,1, <0>
1,854,2,2,345,1,1,810,2,2,169,1
450 DATA1,897,2,2,516,1,1,897,2,2,516,1, <154>
1,864,2,2,383,1,1,897,2,2,516,1
460 DATA1,897,2,2,516,1,1,881,2,2,453,1, <9>
1,864,2,2,383,1,1,864,1,2,383,1
470 DATA1,864,1,2,383,1,1,864,1,2,383,1, <75>
1,897,2,2,516,1,1,881,2,2,453,1
480 DATA1,854,2,2,345,1,1,810,2,2,169,1, <40>
1,854,2,2,345,1,1,810,2,2,169,1
490 DATA1,864,2,2,383,1,-1,0,0 <125>
500 FORI=1T0250:NEXTI:CHAR1,12,22,"(BL.G <101>
RN)DRUECKE(SPACE)">I" <21>
510 GETA$:IFA$="(SPACE)"THEN1030 <29>
520 IFA$="T"THEN1030 <90>
530 GOTO510
540 DATA162,0,189,0,208,157,0,48,189,0,2 <65>
09,157,0,49,202,208,241,96
550 DATA36,24,90,189,219,90,24,0 <63>
560 DATA36,24,90,189,189,189,90,0 <142>
570 DATA153,189,60,126,126,126,189,153 <135>
580 DATA134,168,122,255,255,122,168,134 <41>
590 DATA97,21,94,255,255,94,21,97 <185>
600 DATA153,189,126,126,126,60,189,153 <144>
610 DATA195,195,195,219,219,219,219,255 <165>
620 DATA126,102,96,96,238,230,230,254 <218>
630 DATA0,0,16,38,254,189,68,0,4,15,18,9 <54>
8,127,40,16,0,0,240,88,88
640 DATA248,160,64,0,0,0,248,212,255,189 <195>
,66,0,0,0,56,36,127,221,34,0
650 DATA126,102,102,102,230,230,230,230 <117>
660 DATA17,16,127,117,127,60,24,0,241,14 <71>
5,217,93,223,120,48,0,255,37,37,253,255, <96>
60,24
670 DATA0,0,63,115,94,115,63,0,0,0,251,2 <54>
07,121,207,255,0,0,0,252,62,230,62,252,0
680 DATA65,29,62,254,254,62,29,65,126,12 <233>
9,157,161,161,157,129,126
690 DATA102,102,102,102,254,230,230,230, <205>
126,102,102,102,230,230,230,254
700 DATA0,0,0,42,84,0,0,0,0,255,205,253, <7>
61,255,0,0
710 DATA255,255,255,255,255,255,255,0,0, <54>
0,0,255,255,0,0,0
720 DATA24,24,24,24,24,24,24,24 <113>
730 DATA24,24,24,248,240,0,0,0 <229>
740 DATA0,0,0,240,248,24,24,24 <11>
750 DATA0,0,0,15,31,24,24,24 <15>
760 DATA0,0,0,0,0,0,0,0 <209>
770 DATA24,24,24,31,15,0,0,0 <52>
780 DATA24,24,24,24,24,24,0,0 <86>
790 DATA0,0,0,24,24,24,24,24 <80>
800 DATA24,24,24,255,255,0,0,0 <118>
810 DATA0,0,0,248,248,24,24,24 <162>
820 DATA0,0,0,31,31,24,24,24 <66>
830 DATA0,0,0,255,255,24,24,24 <132>
840 DATA24,24,24,248,248,24,24,24 <120>
850 DATA24,24,24,31,31,24,24,24 <188>
860 DATA0,60,126,255,255,126,60,0 <248>
870 DATA255,255,255,255,255,255,255,255 <88>
880 DATA255,0,255,0,255,0,255,0 <214>
890 DATA126,102,102,102,254,224,224,224 <219>
900 DATA126,96,96,96,248,224,224,254 <17>
910 DATA96,96,96,96,224,224,224,254 <23>
920 DATA96,96,96,96,224,224,224,224 <208>
930 DATA126,24,24,24,24,56,56,56 <83>
940 DATA254,198,192,192,224,224,230,254 <56>
950 DATA254,198,192,254,6,6,230,254 <204>

```



```
960 DATA252,204,204,254,230,230,230,254 <22>
970 DATA252,204,204,254,230,230,230,230 <98>
980 DATA102,102,102,102,230,230,60,24 <14>
990 DATA255,219,219,219,219,195,195,195 <214>
1000 DATA126,102,102,102,254,230,230,230 <74>
1010 DATA102,102,102,126,24,24,24,24 <158>
1020 DATA-1 <111>
1030 L=4:D=6:SC=0 <119>
1040 IFHC<SC THENHC=SC <202>
1050 PRINT" (HOME2 CLEAR RED)" <245>
1060 PRINT" (RVSON SPACE40 RVSOFF)" <205>
1070 PRINT" (RVSON SPACE) + + + + + (SPACE5) 0 (SPACE) X + + + + + (SPACE5) 0 (SPACE) + + + + + (SPACE6 RVSOFF)" <214>
1080 PRINT" *****" <55>
1090 PRINT" (L.BLU RVSON SPACE D.BLUE SPA  
CE3 RVSOFF PINK)-(RVSON D.BLUE SPACE4 RV  
SOFF YL.GRN)* (RVSON D.BLUE SPACE4 RVSOFF  
PINK)-(RVSON D.BLUE SPACE4 RVSOFF YL.GR  
N)* (RVSON D.BLUE SPACE4 RVSOFF PINK)-(RV  
SON D.BLUE SPACE4 RVSOFF YL.GRN)* (RVSON  
D.BLUE SPACE4 RVSOFF PINK)-(RVSON D.BLUE  
SPACE4 L.BLU SPACE RVSOFF)" <38>
1100 FORI=1TO7 <187>
1110 PRINT" (RVSON SPACE RVSOFF BLUE) ****  
***** (L.BLU  
RVSON SPACE RVSOFF)":NEXT <54>
1120 FORI=1TO4 <189>
1130 PRINT" (RVSON SPACE RVSOFF GREEN SPA  
CE38 L.BLU RVSON SPACE RVSOFF)":NEXT <204>
1140 PRINT" (RVSON SPACE RVSOFF YELLOW) ++  
+++++ (L.B  
LU RVSON SPACE RVSOFF)" <111>
1150 FORI=1TO3 <213>
1160 PRINT" (RVSON SPACE RVSOFF L.GRN RIG  
HT38 L.BLU RVSON SPACE RVSOFF)":NEXT <125>
1170 PRINT" (RVSON SPACE RVSOFF YELLOW) ++  
+++++ (L.B  
LU RVSON SPACE RVSOFF)" <141>
1180 PRINT" (RVSON SPACE RVSOFF GREEN RIG  
HT38 L.BLU RVSON SPACE RVSOFF)" <105>
1190 PRINT" (RVSON SPACE40 RVSOFF)" <79>
1200 PRINT" (RVSON SPACE40 RVSOFF)" <89>
1210 CHAR1,22,2," (RVSON L.BLU)":PRINTUSI  
NG"#####":HC <173>
1220 GOSUB1550 <26>
1230 H=1 <17>
```

```
H+4),3B); <203>
1400 IFH=40 THENH=0 <144>
1410 H=H+1:CHAR1,34,2," (RVSON L.BLU)":PR  
INTUSING"#####":S:IFS=<0 THENGOSUB1540 <10>
1420 S=S-50 <22>
1430 CHAR1,7,2," (RVSON L.BLU)":PRINTUSIN  
G"#####":SC <137>
1440 Z=JOY(1) <91>
1450 IFZ=129 THENA=-80:C=66:SOUND2,900,3:  
GOTO1510 <11>
1460 IFZ=133 THENA=+80:C=69:SOUND2,100,3:  
GOTO1510 <111>
1470 IFZ=5 THENA=+40:C=69:SOUND2,300,3:GO  
TO1510 <240>
1480 IFZ=7 THENA=-1:C=68:SOUND2,500,3:GOT  
O1510 <155>
1490 IFZ=1 THENA=-40:C=66:SOUND2,700,3:GO  
TO1510 <31>
1500 IFZ=3 THENA=+1:C=67:SOUND2,500,3 <91>
1510 IFPEEK(B+A)<>32 THENGOTO1720 <102>
1520 POKEB,32:B=B+A:POKEB-1024,31:POKEB,  
C:A=0 <8>
1530 GOTO1320 <59>
1540 L=L-1:POKEB,32:B=3930 <154>
1550 IFL=4 THENGOSUB1620:GOTO1230 <119>
1560 FORA=1TO2:X=1000:FORI=1TO3:X=X-50:S  
OUND2,X,2:SOUND2,X-200,3:SOUND2,X-400,2 <7>
1570 NEXTI:NEXTA <55>
1580 IFL=3 THENGOTO1630 <98>
1590 IFL=2 THENGOTO1640 <113>
1600 IFL=1 THENGOTO1650 <128>
1610 IFL=0 THENGOTO1670 <152>
1620 CHAR1,0,23," (RED) _ _ _ _ _ (GREEN)-(RED  
SPACE GREEN)-(RED SPACE GREEN)-(RED SPA  
CE GREEN)-(RED)":S=2500:RETURN <216>
1630 CHAR1,0,23," (RED) _ _ _ _ _ (GREEN)-(RED  
SPACE GREEN)-(RED SPACE GREEN)-(RED SPA  
CE2)":S=2500:RETURN <224>
1640 CHAR1,0,23," (RED) _ _ _ _ _ (GREEN)-(RED  
SPACE GREEN)-(RED SPACE4)":S=2500:RETUR  
N <224>
1650 CHAR1,0,23," (RED) _ _ _ _ _ (GREEN)-(RED  
SPACE6)":S=2500:RETURN <216>
1660 GOTO1050 <193>
1670 CHAR1,0,23," (RED) _ _ _ _ _ (SPACE7)" <125>
1680 FORI=1TO500:NEXTI <6>
1690 PRINT" (HOME2 RIGHT15 DOWN13) | _ _ _ _ _ (SP  
ACE) 0 _ _ _ _ _ FORI=1TO500:NEXTI=1:GOTO1690 <110>
```





10000:GOTO1900	<230>	2050 IFSC>MDTHENMF\$=ME\$;ME\$=MD\$;MD\$=SC:GOTO2100	<60>
1870 IFK=7ANDM>=500ANDM<1000THENM=M+100:		2060 IFSC>METHENMF\$=ME\$;ME\$=M\$	<4>
SC=SC+10000:K=0:GOTO1900	<253>	2070 IFSC>METHENMF=ME\$;ME\$=SC:GOTO2100	<151>
1880 IFK=7THENK=0:SC=SC+10000:GOTO1900	<244>	2080 IFSC>METHENMF\$=M\$	<248>
1890 GOTO1310	<161>	2090 IFSC>METHENMF=SC	<45>
1900 FORI=1TO5:FORW=1000TO700STEP-25:SOU		2100 PRINT"<CLEAR>":COLOR0,5,7:COLOR4,3,	
ND2,W,1:NEXTW:NEXTI:GOTO1040	<17>	2	<7>
1910 GOTO1310	<181>	2110 PRINT:PRINT:PRINT"<GREEN RIGHT6>-<R	
1920 IFHC<SC THENHC=SC	<61>	IGHT RVSON>FROGGER<SPACE>X<RIGHT1><SPACE>	
1930 RETURN	<31>		<140>
1940 IFSC>METHENPRINT"<CLEAR RED>":CHAR1,		2120 PRINT:PRINT:PRINT	<13>
11,7,"HOECHSTENS<SPACE>>B<SPACE>ZEICHE		2130 COLOR1,15,0:PRINT"<RIGHTS>1:<FLASHO	
N":CHAR1,14,10,"{Y<GRN>\<SPACE>:"	<224>	N}";MA;"-<SPACE>";MA\$	<158>
1950 IFSC<MF THEN1980	<113>	2140 COLOR1,15,1:PRINT"<RIGHTS DOWN>2:"	
1960 FORI=1TO20:GETA\$:NEXT:CHAR1,14,12,"	<179>	MB;"-<SPACE>";MB\$	<1>
<RVSON>":INPUTM\$:IFLEN(M\$)<8THEN1980	<15>	2150 COLOR1,15,2:PRINT"<RIGHTS DOWN>3:"	
1970 GOTO1940		MC;"-<SPACE>";MC\$	<96>
1980 IFSC>MATHENMF\$=ME\$;ME\$=MD\$;MD\$=MC\$:		2160 COLOR1,15,4:PRINT"<RIGHTS DOWN>4:"	
MC\$=MB\$;MB\$=MA\$;MA\$=M\$	<167>	MD;"-<SPACE>";MD\$	<198>
1990 IFSC>MATHENMF=ME\$;ME\$=MD\$;MD\$=MC\$:		2170 COLOR1,15,5:PRINT"<RIGHTS DOWN>5:"	
MB=MA\$;MA\$=SC:GOTO2100	<144>	ME;"-<SPACE>";ME\$	<37>
2000 IFSC>MBTHENMF\$=ME\$;ME\$=MD\$;MD\$=MC\$:		2180 COLOR1,15,6:PRINT"<RIGHTS DOWN>6:"	
MC\$=MB\$;MB\$=M\$	<19>	MF;"-<SPACE>";MF\$	<132>
2010 IFSC>MBTHENMF=ME\$;ME\$=MD\$;MD\$=MC\$:		2190 PRINT:PRINT:PRINT"<CYAN RIGHT	
MB=SC:GOTO2100	<208>	12>DRUECKE<SPACE>>X<RIGHT1><	<118>
2020 IFSC>MCTHENMF\$=ME\$;ME\$=MD\$;MD\$=MC\$:		2200 GETA\$:IFA\$="<SPACE>" THENCOLOR4,1:CO	
MC\$=M\$	<177>	LOR0,1:RETURN	<174>
2030 IFSC>MCTHENMF=ME\$;ME\$=MD\$;MD\$=MC\$:	<1>	2210 GOTO2200	<221>
GOTO2100			
2040 IFSC>MDTHENMF\$=ME\$;ME\$=MD\$;MD\$=M\$	<161>		

## Das der C-16 „im Kommen ist“ beweist nicht zuletzt dieses Programm

## Aktion fehlerhaftes Listing . . . Aktion fehlerhaftes Listing . . .

Teilnehmen an unserer Aktion „fehlerhaftes Listing“ können wirklich nur die eingesandten Coupons.

Ein Hinweis: Das Listing ist nur dann fehlerhaft, wenn das Programm sich aufhängt bzw. eine ERROR-Meldung ausgibt (richtige Programmier eingabe ist natürlich Voraussetzung).

**Die 5 Gewinner erhalten 100,-- DM!**

**Hinweis:** In der Computronic-Nr. 3 ist uns kein Fehler unterlaufen, ein Pluspunkt für Redaktion und Programmierabteilung. Aber vielleicht können Sie uns dieses Mal ein Fehler nachweisen!?

# HUMOR

*Humor müßte man haben. Der Fremde, der in Ihrem Haus war, hat jedenfalls keinen. Denn während Sie weg waren, hat er 9 Bomben in Ihrem „Domizil“ versteckt.*

Ihre gefährliche Aufgabe ist es nun, diese Bomben aufzusammeln und aus dem Haus zu werfen. Allerdings werden Sie dabei von drei Monstern gestört. Sollten Sie diese berühren, so bedeutet das Ihr vorzeitiges Ende. Das Spiel ist auch verloren, wenn Sie Ihre Aufgabe nicht in der vorgegebenen Zeit geschafft haben.

Vorsicht! In dem Haus sind noch zwei Falltüren versteckt. Tappt man in diese Fallen herein, so kostet das zwar kein Computereleben, aber sehr viel Zeit. Auch wenn Sie die Treppen herunterstolpern, kostet das wertvolle Zeitpunkte. Für jede aufgenommene Bombe bekommen Sie allerdings 10 Punkte gutgeschrieben. Weitere 100 Punkte erhalten Sie, wenn Sie einen Level geschafft haben.

Haben Sie alle Bomben eingesammelt, müssen Sie diese aus dem Fenster werfen. Nun werden Ihnen weitere 100 Punkte gutgeschrieben. Um von einer Etage zur anderen zu gelangen, können Sie die Treppen und die Fahrstühle benutzen.

**Spaß - Spannung -  
Nervenkitzel**



## Humor: Das Listing

1 REM *****	<106>	255	<35>
2 REM *** HUMOR ***	<80>	34 DATA 113,255,195,165,153,153,165,195,	<0>
3 REM *** WRITTEN BY ***	<73>	255	
4 REM *** BJOERN THIEMANN ***	<222>	35 DATA 79,255,130,132,136,144,160,192,1	
5 REM *** 3014 LAATZEN 1 ***	<235>	28	<99>
6 REM *** TEL 0511/826459 ***	<88>	36 DATA 42,60,90,255,189,195,255,36,102	<72>
7 REM *****	<112>	37 DATA 44,126,153,153,231,102,60,36,24	<115>
8 REM	<151>	38 DATA 46,60,90,255,195,126,36,36,231	<177>
9 TRAP 254	<179>	39 CLR	<195>
10 COLOR 4,15,2:COLOR 0,1	<15>	40 RESTORE 255	<139>
11 REM*****	<184>	41 REM*****	<62>
12 REM* ZEICHEN DEFINIEREN *	<195>	42 REM* EINLEITUNG *	<143>
13 REM*****	<186>	43 REM*****	<64>
14 POKE 51,0:POKE 52,60:POKE 55,0:POKE 5		44 SCNCLR	<20>
6,50:POKE 65298,0:POKE 65299,60	<161>	45 CHAR 1,7,1,"(BLUE) (SPACE) "	<110>
15 FOR T=15300 TO 15333	<215>	46 CHAR 1,7,2," (SHIFTSPACE) "	<48>
16 READ A1:POKE T,A1:NEXT	<78>	47 CHAR 1,7,3,"   (SPACE) (SPACE2	
17 DATA 169,000,133,003,133,005,169,211,		)  (SPACE) "	<108>
133,004,169,063,133,006,160,000,177	<172>	48 CHAR 1,7,4,"   (SHIFTSPACE)	
18 DATA 003,145,005,136,208,249,198,004,		"	<41>
198,006,165,006,201,059,208,237,096	<187>	49 CHAR 1,7,5,"   (SPACE)	
19 SYS 15300	<142>	"	<197>
20 FOR T=0 TO 15:READ ZN:X=1	<24>	50 CHAR 1,7,6,"  (SPACE)         (SPACE2	
21 V1=15360+8*ZN:FOR C1=0 TO 7	<43>	)         "	<200>
22 READ H1:POKE V1+C1,H1:NEXT:NEXT	<88>	51 CHAR 1,0,8,"(SPACE3 D.BLUE)COPYRIGHT (	
23 DATA 102,236,236,0,118,118,0,236,236	<42>	SPACE) 1986 (SPACE) BY (SPACE) BJOERN (SPACE) T	
24 DATA 81,8,8,8,28,62,62,62,28	<224>	HIEMANN"	<104>
25 DATA 87,56,100,255,240,124,112,80,80	<140>	52 CHAR 1,1,10,"(GREEN)LEGENDE: (SPACE2 D	
26 DATA 86,88,68,120,112,96,96,96,88	<23>	RNG)-+ (SPACE) FAHRSTUHL (SPACE) NACH (SPACE)	
27 DATA 90,28,38,255,15,62,14,10,10	<122>	OREN"	<76>
28 DATA 88,26,34,30,14,6,6,6,30	<143>	53 CHAR 1,1,11,"(SPACE10 PINK)-+ (SPACE) F	
29 DATA 65,60,90,126,36,24,24,255,189	<106>	AHRSTUHL (SPACE) NACH (SPACE) UNTEN"	<208>
30 DATA 83,189,189,60,60,24,24,24,126	<54>	54 CHAR 1,1,12,"(SPACE10 BL.GRN)-+ (SPACE	
31 DATA 80,255,65,33,17,9,5,3,1	<203>	) TREPPE (SPACE) NACH (SPACE) LINKS"	<149>
32 DATA 114,255,129,161,161,161,189,129,		55 CHAR 1,1,13,"(SPACE10 PURPLE)-+ (SPACE	
255	<196>	) TREPPE (SPACE) NACH (SPACE) RECHTS"	<153>
33 DATA 120,255,169,131,209,139,193,149,		56 CHAR 1,1,14,"(SPACE10 YELLOW)-+ (SPACE	
		) FALLTUE"	<85>



```

57 CHAR 1,1,15,"{SPACE10 L.GRN}-●{SPACE} <162>
BOMBE"
58 CHAR 1,1,16,"{SPACE10 BRN}-|{SPACE3FE <136>
NSTER"
59 RESTORE 259 <174>
60 READ KA,S1:IF S1=0 THEN GOTO 57 <222>
61 GET AB$:IF AB$="{SPACE}" THEN 67 <149>
62 CHAR 1,1,18,"BITTE{SPACE}DRUECKEN{SPA <78>
CE}SIE{SPACE}SPACE"
63 FOR T=8 TO 0 STEP -1 <98>
64 VOL T:SOUND 1,KA,S1 <208>
65 NEXT <195>
66 GOTO 60 <199>
67 PRINT:PRINT <27>
68 CLR <224>
69 INPUT "{RED SPACE}SOUND{SPACE}ON/OFF{S <112>
PACE}(1/0)";MU
70 CHAR 1,1,22,"{FLASHON WHITE}BITTE{SPA <50>
CE}JOYSTICK{SPACE}BEWEGEN{FLASHOFF}"
71 GETKEY JA$ <112>
72 IF JOY(1)>0 THEN CR=1 <169>
73 IF JOY(2)>0 THEN CR=2 <186>
74 IF CR=0 THEN CR=1 <132>
75 FL=10:TM=-500:BO=9:LE=01:M=1 <113>
76 VOL 8 <151>
77 REM***** <236>
78 REM* SPIELBILD * <6>
79 REM***** <238>
80 SCNCLR <56>
81 CHAR 1,10,0,"{SPACE5}-{FLASHON SPACE} <105>
HUMOR{SPACE FLASHOFF}-{SPACE7}"
82 CHAR 1,0,1,"{RED} <230>
"
83 CHAR 1,0,2,"{SPACE12}{SPACE12}{SPA <211>
CE12}"
84 CHAR 1,0,3,"{SPACE12}{SPACE12}{SPA <218>
CE12}"
85 CHAR 1,0,4,"{SPACE12}{SPACE12}{SPA <225>
CE12}"
86 CHAR 1,0,5,"{SPACE12}{SPACE8 GREEN} <2>
●{RED SPACE3}{SPACE12}"
87 CHAR 1,0,6,"{BRN}|{RED SPACE20 PURPLE <162>
3}{ORNG}{SPACE15 BRN}|"
88 CHAR 1,0,7,"{SPACE5 WHITE}●{SPACE13 <149>
}{SPACE3 BLUE}{SPACE7 CYAN}●{SPACE6 BRN
3}{RED}"
89 CHAR 1,0,8,"{GREEN}+{RED} <233>
{WHITE}{RED SPACE3}
"
90 CHAR 1,0,9,"{SPACE12}{SPACE}{SPACE <132>
10}{SPACE12}"
91 CHAR 1,0,10,"{SPACE12}{SPACE2 CYAN} <175>
}{RED SPACE9}{SPACE12}"
92 CHAR 1,0,11,"{SPACE12}{SPACE3 PURPL <145>
E}{RED SPACE8}{SPACE12}"
93 CHAR 1,0,12,"{SPACE12}{SPACE4 GREEN <249>
}{RED SPACE7}{SPACE12}"
94 CHAR 1,0,13,"{SPACE18 BLUE}{RED SPA <218>
CE19}"
95 CHAR 1,0,14,"{SPACE8 PURPLE}●{SPACE6 <10>
GREEN}●{SPACE2 YELLOW SPACE}{BRN SPACE
2 BLUE}●{SPACE5 RED SPACE3}"
96 CHAR 1,0,15,"{GREEN}+{RED}{SPACE3} <72>
{SPACE3 BLUE}{GREEN}+{RED}"
97 CHAR 1,0,16,"{SPACE3 ORNG}{RED SPAC <87>
E8}{SPACE10 L.GRN}{RED SPACE}{SPACE2
YELLOW}{RED SPACE2}"
98 CHAR 1,0,17,"{SPACE4 BRN}{RED SPACE <58>
7}{SPACE9 WHITE}{RED SPACE2}{SPACE8 O
RNG}{RED SPACE3}"
99 CHAR 1,0,18,"{SPACE5 YL.GRN}{RED SP <210>
ACE6}{SPACE8}{SPACE3}{SPACE7 BRN}{RE
D SPACE4}"
100 CHAR 1,0,19,"{SPACE6 PINK}{RED SPA

```

```

CE5}{SPACE7 CYAN}{RED SPACE4}{SPACE6 <43>
YL.GRN}{RED SPACE5}"
101 CHAR 1,0,20,"{SPACE7 BL.GRN}{RED S <165>
PACE11 PURPLE}{RED SPACE11 PINK}{RED S
PACE6}"
102 CHAR 1,0,21,"{SPACE3 YELLOW}●{SPACE <126>
4 L.BLU}{RED SPACE6 ORNG}●{SPACE2 GREEN
3}{RED SPACE3 YL.GRN}●{SPACE7 GREEN}{RE
D SPACE7}"
103 CHAR 1,0,22,"{GREEN}+{RED} <35>
"
104 CHAR 1,0,23,"{SPACE2 PINK}TIME:{SPA <255>
CE7}LEVEL:{SPACE5}SCORE:{SPACE7 RED}"
105 CHAR 1,0,24," <217>
"
106 POKE 4071,102 <107>
107 IF NM=1 THEN RETURN <2>
108 REM***** <47>
109 REM* VARIABLEN/JOYSTICK/SOUND * <113>
110 REM***** <49>
111 X=1:Y=21:B=0:BW=1:RESTORE 255 <72>
112 J=JOY(CR) <133>
113 IF J=3 THEN GOSUB 165:BW=1 <205>
114 IF J=7 THEN GOSUB 172:BW=2 <217>
115 TM=TM+1:IF TM=0 THEN 215 <82>
116 T$=STR$(TM):CHAR 1,8,23,T$+"{SPACE}" <12>
117 L$=STR$(LE):CHAR 1,21,23,L$+"{SPACE} <168>
"
118 SC$=STR$(SC):CHAR 1,32,23,SC$+"{SPAC <66>
E}"
119 IF MU=0 THEN GOTO 124 <78>
120 READ TE,LA <40>
121 FOR VO=3 TO 0 STEP-1 <183>
122 VOL VO:SOUND 1,TE,LA:NEXT:VOL 8 <99>
123 IF TE=0 THEN RESTORE 255 <114>
124 REM***** <99>
125 REM* FALLTUE/ABFRAGEN * <207>
126 REM***** <101>
127 CHAR 1,31,8,"{SPACE1,22,6," <98>
128 FA=INT(RND(1)*FL)+1:IF FA=1 THEN O=1 <106>
129 IF O=1 THEN CHAR 1,31,8,"{SPACE3}":C <52>
HAR 1,22,6,"{SPACE}":O=0
130 IF PEEK(3072+X+40*(Y+1))<>32 THEN F= <233>
0
131 IF PEEK(3072+X+40*Y)=79 OR PEEK(3072 <185>
+X+40*Y)=80 THEN Y=Y-1
132 IF PEEK(3072+X+40*Y)=81 THEN B=B+1:S <237>
C=SC+10:VOL B:SOUND 1,900,3
133 IF PEEK(3072+X+40*(Y-1))=79 OR PEEK( <233>
3072+X+40*(Y-1))=80 THEN GOSUB 215
134 IF PEEK(3072+X+40*(Y+1))=32 THEN GOS <232>
UB 179
135 IF PEEK(3072+X+40*(Y+1))=114 AND J=1 <7>
28 THEN GOSUB 186:TM=TM+10
136 IF PEEK(3072+X+40*(Y+1))=113 AND J=1 <95>
28 THEN GOSUB 201:TM=TM+10
137 IF PEEK(3072+X+40*Y)=66 THEN 231 <49>
138 REM***** <243>
139 REM* LAUFRICHTUNG BESTIMMEN * <253>
140 REM***** <245>
141 IF BW=1 THEN CHAR 1,X,Y-1,"{PINK} <184>
"
142 IF BW=1 THEN CHAR 1,X,Y,"{BLUE}X" <136>
"
143 IF BW=2 THEN CHAR 1,X,Y-1,"{PINK}+ <245>
"
144 IF BW=2 THEN CHAR 1,X,Y,"{BLUE}+ <175>
"
145 REM***** <84>
146 REM* MONSTER:ABFRAGEN/BEWEGEN * <161>
147 REM***** <86>
148 LO=PEEK(3072+M+1+40*7) <117>
149 ES=PEEK(3072+M+1+40*14) <16>
150 CH=PEEK(3072+M+1+40*21) <210>
151 IF LO=88 OR LO=86 THEN 231 <246>
152 IF ES=88 OR ES=86 THEN 231 <225>
153 IF CH=88 OR CH=86 THEN 231 <74>
154 M=M+1:IF M=39 THEN M=0 <92>
155 POKE 3072+M+40*7,42 <19>

```





```

156 POKE 3072+M+40*14,44
157 POKE 3072+M+40*21,46
158 POKE 3072+M+40*7,LO
159 POKE 3072+M+40*14,ES
160 POKE 3072+M+40*21,CH
161 GOTO 112
162 REM** *****
163 REM* MAENCHEN BEWEGEN(RECHTS) *
164 REM** *****
165 CHAR 1,X,Y-1,"{SPACE}"
166 CHAR 1,X,Y,"{SPACE}"
167 X=X+1:IF PEEK(3072+X+40*Y)=102 THEN
X=X-1
168 RETURN
169 REM*****
170 REM* MAENCHEN BEWEGEN(LINKS) *
171 REM*****
172 CHAR 1,X,Y-1,"{SPACE}"
173 CHAR 1,X,Y,"{SPACE}"
174 X=X-1:IF PEEK(3072+X+40*Y)=102 THEN
X=X+1
175 RETURN
176 REM*****
177 REM* MAENCHEN FALLEN *
178 REM*****
179 CHAR 1,X,Y-1,"{SPACE}"
180 CHAR 1,X,Y,"{SPACE}"
181 Y=Y+1:F=F+1:IF F>3 THEN SC=SC-5
182 RETURN
183 REM*****
184 REM* FAHRSTUHL OBEN *
185 REM*****
186 FOR I=1 TO 7
187 CHAR 1,X,Y-1,"{PINK}▲"
188 CHAR 1,X,Y,"{BLUE}▼"
189 CHAR 1,X,Y+1,"{GREEN}┐"
190 FOR T=1 TO 100:NEXT T
191 CHAR 1,X,Y+1,"{SPACE}"
192 Y=Y-1:SOUND 3,L*100,10
193 NEXT I:BW=1
194 CHAR 1,X,Y+1,"☼"
195 CHAR 1,X,I+7,"┘"
196 CHAR 1,2,22,"┘"
197 RETURN
198 REM*****
199 REM* FAHRSTUHL UNTEN *
200 REM*****
201 FOR I=1 TO 7
202 CHAR 1,X,Y-1,"{PINK}▲"
203 CHAR 1,X,Y,"{BLUE}▼"
204 CHAR 1,X,Y+1,"{GREEN}┐"
205 FOR T=1 TO 100:NEXT T
206 CHAR 1,X,Y-1,"{SPACE}"
207 Y=Y+1:SOUND 3,L*100,10
208 NEXT I:BW=1
209 CHAR 1,X,Y+1,"☼"
210 CHAR 1,11,8,"┘"
211 RETURN
212 REM*****
213 REM* GAME OVER *
214 REM*****
215 FOR T=3072 TO 4071
216 IF PEEK(T)=81 THEN 217:ELSE NEXT:GOT

```

## HUMORLISTING ENDE

```

0 219
217 POKE T+40,32:POKE T,32
218 FOR V=8 TO 0 STEP -1:VOL V:SOUND 3,9
30,3:NEXT:NEXT
219 COLOR 0,2,7:COLOR 0,1
220 COLOR 0,2,7
221 SCNCLR
222 FOR T=1023 TO 0 STEP -1:VOL 8:SOUND
1,T,1:NEXT
223 CHAR 1,5,11,"{SPACE2}"
224 CHAR 1,5,12,"{SPACE4}"
225 CHAR 1,5,13,"{SPACE2}"
226 FOR T=1 TO 3000:NEXT
227 COLOR 0,1:RUN
228 REM*****
229 REM* LEVEL BEWAELTIGT *
230 REM*****
231 IF B<>BO THEN 215
232 NM=1
233 SOUND 1,900,10:SOUND 1,800,10:M=1
234 CHAR 1,12,0,"{FLASHON}LEVEL{SPACE}"+
L+="{SPACE}BEWAELTIGT{FLASHOFF}"
235 FOR T=1 TO 3000:NEXT
236 ON LE GOTO 237,241,245,249
237 LE=LE+1:SC=SC+100:TM=-450:BO=BO+3
238 GOSUB 81
239 CHAR 1,3,7,"{PINK}●":CHAR 1,3,14,"{B
RN}●":CHAR 1,7,21,"{BL.GRN}●"
240 GOTO 111
241 LE=LE+1:SC=SC+100:TM=-400:FL=6
242 GOSUB 81
243 CHAR 1,19,7,"{L.BLUE}●":CHAR 1,27,14,
"{PINK}●":CHAR 1,12,21,"{D.BLUE}●"
244 GOTO 111
245 LE=LE+1:SC=SC+100:TM=-350:FL=3
246 GOSUB 81
247 CHAR 1,37,7,"{L.GRN}●":CHAR 1,33,14,
"{L.BLUE}●":CHAR 1,28,21,"{PURPLE}●"
248 GOTO 111
249 LE=4:ZE=ZE-50:TM=-350+ZE
250 GOTO 245
251 REM*****
252 REM* DATAS FUER SOUND *
253 REM*****
254 SYS 65529
255 DATA 917,1,917,1,939,2,917,1,939,1,9
39,1,953,1,939,1,924,2,929,2,917,1,929,1
256 DATA 939,2,944,1,953,1,939,1,929,1,9
17,1,939,2,929,1,944,2,939,1,929,1,917,1
,911,1
257 DATA 929,2,944,2,939,1,929,1,917,1,9
29,1,911,1,929,1,929,1,944,1,939,1,917,1
258 DATA 939,2,929,2,0,0
259 DATA 810,1,834,1,854,1,854,1,854,1,8
54,2,834,1,834,1,854,1,854,1,854,2
260 DATA 854,1,881,1,834,1,834,1,834,1,0
,0
261 REM*****
262 REM* ALLERHOEHSTES ENDE *
263 REM*****

```

### IMPRESSUM

Computronic  
erscheint alle 2 Monate im  
Tronic-Verlag, Postfach 870  
3440 Eschwege, Tel. 05651/30011

Herausgeber: Axel Crede  
Chefredakteur: Siegfried Gork  
Redaktion: Manfred Kleimann, Hartmut Wendt,  
Frank Brall, Otfried Schmidt, Bernd Zimmermann

Programmeautoren: O. Schmidt, T. Goessmann, O.  
Cyranka, O. Steinmeier, K. Ezcan, F. Brall, W. Bem-  
bon, Z. Urwani, G. Stöhr und einige andere.

Gesamtherstellung:  
Druckhaus Dierichs Kassel, Frankfurter Str. 168,  
3500 Kassel

Vertrieb:  
Inland (Groß-, Einzel- und Buchhandelsbuchhandel)  
sowie Österreich und Schweiz:  
Verlagsunion  
Friedrich-Bergius-Straße 20  
6200 Wiesbaden  
Telefon 06121/2660

Anzeigenverwaltung (Inland):  
Hartmut Wendt, Tronic-Verlag, Postfach 870,  
3440 Eschwege, Tel. 05651/30011

Anzeigenleitung (Ausland):  
Public Relation (Foreign Countries):  
Hartmut Wendt, c/o Tronic-Verlag.



Postfach 870 (Postbox 870),  
D-3440 Eschwege, Tel. 05651/30011

Erscheinungsweise: Erstverkaufstag von  
COMPUTRONIC ist Anfang des Monats

Anzeigenpreisliste:  
Bitte Medienunterlagen anfordern.

Software-Service und Vertrieb: Annelie Kratzberg,  
Heike Rabe.

Urheberrecht:  
Alle in COMPUTRONIC veröffentlichten Beiträge  
sind urheberrechtlich geschützt. Alle Rechte, auch  
Übersetzungen, vorbehalten.  
Reproduktionen jeder Art (Fotokopiert, Microfilm,  
Erfassung in Datenverarbeitungsanlagen usw.) be-

dürfen der schriftlichen Genehmigung des Verlags.  
Alle veröffentlichte Software wurde von Mitarbei-  
tern des Verlages oder von freien Mitarbeitern  
erstellt.  
Aus Ihrer Veröffentlichung kann nicht geschlossen  
werden, daß die beschriebenen Lösungen oder  
Bezeichnungen frei von Schutzrechten sind.

Bezugspreise:  
Einzelheft 6,50 DM

Autoren, Manuskripte:  
Der Verlag nimmt Manuskripte und Software zur  
Veröffentlichung gerne entgegen. Sollte keine an-  
dere Vereinbarung getroffen sein, so gehen wir da-  
von aus, daß Sie mit einem Honorar von 120,- DM  
pro abgedruckter Seite im Heft einverstanden sind.

Titelblatt: Werbeberater H. Kästle, Eschwege

ISSN-0179-6739

# Traumland ist das erste Textadventure, das wir in der **COMPUTRONIC** veröffentlichen. Dieser Programmbeitrag wird Ihnen mit Sicherheit viel Spaß und Freude bereiten.

## TRAUMLAND - ein deutschsprachiges Adventure auf dem Schneider CPC-464

Traumland ist ein reines Textadventure. Sie müssen im »Traumland« Schätze sammeln, um möglichst viele Punkte zu bekommen. Doch oft liegen diese im Verborgenen oder sind von Wächtern bewacht, die nur mit bestimmten Formeln oder Werkzeugen zu beseitigen sind. Sie müssen alle Wörter, außer den Richtungsangaben, ausschreiben. Wollen Sie z.B. nach Süden, tippen Sie einfach „S“ ein. Außerdem bestehen die Anweisungen meist aus Zweiwortformen (Verb-Nomen), wie z. B. »stich Baer«.

### Zur Eingabe:

Zuerst tippen Sie den Loader ein und speichern ihn ab. Dann den 2. Loader und abspeichern. Zuletzt das Hauptprogramm eintippen und abspeichern. Spulen Sie die Cassette zurück und geben Sie RUN und ENTER ein. Nun wird der Loader mit Copyrightvermerk geladen, an-



„Also, wer möchte als erster auf die Erde zurück, um das Schneiderprogramm „Traumland“ zu testen?“

schließend eine Musik und dann das Hauptprogramm. Nachdem die 2. Musik ertönt ist, drücken Sie bitte die Taste 1 für neues Spiel. Haben

Sie schon mal einen Spielstand abgespeichert, können Sie Taste 2 drücken.

**Tip zu Spiel:** Es ist wichtig, sich die Notizen und Schriften genau durchzulesen . . .

```

10 '*****
20 '*      *
30 '* Loader von *
40 '*      *
50 '* TRAUMLAND *
60 '*      *
70 '* (c) JR-SOFT *
80 '*      *
90 '*****
100 MODE 0
110 SYMBOL AFTER 32
120 M$="J R - SOFT":PEN 1:LOCATE 6,3:GOS
UB 220:M$="presents":PEN 2:LOCATE 7,10:G
OSUB 220
130 M$=T$:PEN 3:LOCATE 10-(LEN(M$)/2),19
:GOSUB 220
140 LOCATE 4,16:PRINT CHR$(150);STRING$(
12,154);CHR$(156);
150 LOCATE 4,17:PRINT CHR$(149);STRING$(
12,32 );CHR$(149);
160 LOCATE 4,18:PRINT CHR$(149);STRING$(
12,32 );CHR$(149);
170 LOCATE 4,19:PRINT CHR$(147);STRING$(
12,154);CHR$(153);
180 m$=" TRAUMLAND ":LOCATE 5,17:PEN 1:
GOSUB 220
190 PEN 4
200 LOCATE 3,24:PRINT"Bitte warten...."
210 RUN"!traumland"

```

```

220 DATA 8,232,234,266,268,300,302,334,3
36,416,418,498,500,548,550,630
230 matr=PEEK(&B296)+256*PEEK(&B297)
240 FOR N=1 TO LEN(M$)
250 FOR T=0 TO 7
260 ADR=PEEK(MATR+(ASC(MID$(M$,N,1))-32)
*8+T)
270 A(T)=ADR:NEXT:SYMBOL 254,A(0),A(0),A
(1),A(1),A(2),A(2),A(3),A(3):SYMBOL 255,
A(4),A(4),A(5),A(5),A(6),A(6),A(7),A(7)
280 PRINT CHR$(254);CHR$(10);CHR$(8);CHR
$(255);CHR$(11);:SOUND 1,1000,10,7
290 NEXT
300 RETURN

```

### Prüfsumme

ZEILENNR.:	SUMMEN	190-	200:	26223
10-	20:	9548	210-	220: 43567
30-	40:	12529	230-	240: 53584
50-	60:	10115	250-	260: 35700
70-	80:	10640	270-	280: 68501
90-	100:	8454	290-	300: 377
110-	120:	12664	-----	
130-	140:	97486	GESAMTCHECKSUMME:	
150-	160:	77578		
170-	180:	79542		
				<u>2220</u>



```

10 ' Musik zu TRAUMLAND
20 ' Listing 2
30 '
40 DATA 1,1136,23,33,1136,48,12,758,75,1
,1012,25,33,956,50,12,638,75,1,758,25,33
,851,50,12,676,75,1,1136,25,33,956,50,12
,716,75,1,568,23,1,0,4,33,568,46,12,758,
75
50 DATA 1,0,4,1,568,23,33,638,37,12,956,
75,1,758,13,1,851,25,33,758,75,12,1012,7
5,1,0,2,33,758,48,12,1012,50,1,568,23,1,
0,4,33,568,46,12,758,75,1,0,4,1,568,23,3
3,638,23,12,956,75,1,0,4,1,638,23
60 DATA 1,758,25,33,851,24,12,676,75,1,0
,2,1,851,12,1,0,2,1,851,24,1,1136,12,33,
956,50,12,716,75,1,1136,23,1,0,3,33,1136
,50,12,758,75,1,956,12,1,851,13,33,758,4
8,12,1012,75,1,0,4,1,851,13,1,956,12,33,
1136,100,12,758,100,0,0,0
70 ON SQ(1) GOSUB 90
80 GOTO 70
90 READ k,t,1:IF 1=0 THEN FOR i=1 TO 200
0:NEXT:GOTO 100 ELSE SOUND k,t/4,1*1.5:R
ETURN
100 RUN"!traumland2"

```

## Prüfsumme

ZEILENNR.: SUMMEN

10-	20:	23717
30-	40:	18181
50-	60:	74460
70-	80:	3004
90-	100:	48618

Gesamtchecksumme  
36908

```

10 RANDOMIZE TIME
20 oa=1:MODE 1:INK 0,19:INK 1,0:INK 2,26
:INK 3,8:PEN 1
30 BORDER 13:PAPER 0:CLS:LOCATE 14,2:PRI
NT CHR$(150);:FOR i=1 TO 12
40 PRINT CHR$(154);:NEXT:PRINT CHR$(156)
;
50 PRINT TAB(14);CHR$(149);SPC(12);CHR$(
149):PRINT TAB(14);CHR$(149);" TRAUM-LAN
D ";CHR$(149):PRINT TAB(14);CHR$(149);SP
C(12);CHR$(149):PRINT TAB(14);CHR$(147);
:FOR i=1 TO 12
60 PRINT CHR$(154);:NEXT:PRINT CHR$(153)
70 DIM o$(51),r(51,3),ge(44),we(44),ge$(
44),no$(44),ve$(31),r$(3),a$(3)
75 '
80 ' ORTE
85 '
90 RESTORE:FOR i=1 TO 40:READ o$(i):NEXT
100 FOR i=41 TO 50:o$(i)="In einem unhei
mlichen Wald. "+CHR$(24)+"Ma
n sieht kaum einen Weg."+CHR$(24)+" Ob h
ier ein Drache haust ?!":NEXT
110 o$(51)="Auf einer unzugänglichen Wa
ldlichtung."
120 DATA vor einem riesigen Torbogen.,au
f einer Blumenwiese.,an einem Seerosente

```

ich.,in einer trostlosen Einoede.,in ein  
em nebligen Moor.,in einer feenhaften Gr  
otte.,an einer zerklüfteten Felswand.,i  
n einem lichten Auwald.,auf einer bluehe  
nden Heide.

130 DATA in einer verdorrten Steppe.,auf  
dem Versammlungsplatz der Gnome.,in der  
Waffenkammer der Gnome.,in einer Tropfs  
teinhöhle.,am Fusse eines Berges.,in de  
r Schatzkammer der Elfen.,auf einer Berg  
wiese.,an einer silbernen Quelle.

140 DATA in einer Schlucht mit vielen Fa  
rnen.,in einer Baerenhöhle.,auf einer s  
chwankenden Haengebrücke.,an einem Wies  
enhang.,in einer Rosenlaube.,in einem Kr  
äutergarten.,auf einem verschlungenen W  
eg.,an einer verfallenen Burgruine.

150 DATA in einem sonnigem Obsthain.,in  
einem verwilderten Park.,in einer finste  
ren Hexenküche.,im Hof einer verlassen  
en Burg.,in einem finsternen Burgverlies.,  
m,in einem feuchten Laubwald.,im Reich d  
er Feen.

160 DATA in einer grauen Steinwüste.,m,  
auf einem dreckigen Pfad.,m,m,m,in einem  
Hohlweg.

170 o\$(31)="in der Krone eines Baumes. "  
+CHR\$(24)+"Ich kann nicht weit sehen  
." +CHR\$(24)

180 o\$(37)="in einem Dornengebüsch. "+C  
HR\$(24)+"Ich finde mich nicht mehr zurec  
ht." +CHR\$(24):o\$(38)=o\$(87):o\$(39)=o\$(37  
)

190 o\$(35)="an einem breiten Fluss. "+CH  
R\$(24)+"Am anderen Ufer bewegt sich etwa  
s." +CHR\$(24)

195 '  
200 ' RICHTUNGEN  
205 '

210 FOR i=1 TO 51:FOR j=0 TO 3:READ r(i,  
j):NEXT:NEXT

220 DATA 0,0,2,0,8,0,3,1,9,0,0,2,10,0,5,  
0,11,0,0,4,13,0,7,0,0,8,0,0,7,2,9,0,15,3  
10,8,0,4,11,9,0,5,12,10,0,0,0,11

230 DATA 16,6,0,0,17,0,15,0,0,9,0,14,0,1  
3,17,0,0,14,18,16,24,0,19,17,25,0,20,18,  
0,0,21,19,26,0,0,20,27,0,23,0,0,0,24,22,  
29,18,0,23

240 DATA 30,19,0,0,32,21,0,0,0,22,28,0,0  
0,0,27,0,0,30,0,0,25,0,0,0,0,0,0,26,0  
33,0,34,32,0,33,36,37,35,0,0,34,0,34,0,  
5,0

250 DATA 37,38,39,34,37,38,38,38,40,39,3  
9,37,42,39,0,0,43,41,42,41,44,40,42,41,4  
7,41,43,43,48,42,45,44,49,45,45,44,50,51  
0,0,47,43,48,47,48,44,48,47

260 DATA 0,45,50,0,0,46,0,49,46,0,0,0  
265 '

270 ' NOMEN  
275 '

280 FOR i=1 TO 44:READ no\$(i):NEXT

290 DATA N,S,O,W,Norden,Sueden,Osten,Wes  
ten,Zauberbraten,Raupe,Buch,Zettel,Tarnk  
appe,Riesenfrucht,Feuerschwert,Baer,Schi  
ld,Zwerg,Dracheneier,Stein,Höhlenmalere  
i,Strauch,Beeren,Blaetter,Diamantmesser,  
Silberschlüssel





```

300 DATA Baum,Pergament,Steinschleuder,W
asserkrug,Zauberschuhe,Schrifttafel,Zwer
gendolch,Kugel,Kaestchen,Gnom,Ring,Brief
,Rubin,Tuer,Feenhaar,Drache,Holztafel,Po
kal
305 '
310 ' VERBEN
315 '
320 FOR i=1 TO 31:READ ve$(i):NEXT
330 DATA gehe,stich,sieh,inventar,save,l
oad,traumland,ende,wirf,toete,lies,fuell
e,oeffne
340 DATA benuetze,simsalabim,sage,giesse
,schiesse,fuettere,schneide,sperre,hilfe
,klettere,schleudere
350 DATA iss,nimm,verjage,lege,drehe,spr
inge,trinke
355 '
360 ' ANTWORTEN
365 '
370 a$(0)="Sei nicht albern."
380 a$(1)="Ich habe nicht das richtige W
erkzeug."
390 a$(2)="Ich weiss nicht wie das geht.
"
400 a$(3)=" laesst mich nicht."
405 '
410 ' GEGENSTAENDE
415 '
420 FOR i=9 TO 39:READ ge$(i):NEXT
430 DATA einen Zauberkuchen,eine haarige
Raupe,ein gruenes Buch,einen Zettel,ein
e Tarnkappe,eine Riesenfrucht,ein Feuers
chwert,einen grossen Baeren,ein Schild a
us Bronze,einen Zwerg,goldene Dracheneie
r,einen Stein,eine Hoehlenmalerei
440 DATA einen haesslichen Strauch,viele
rote Beeren,goldene Blaetter,ein Diaman
tmesser,einen Silberschuessel,einen due
rren Baum,ein vergilbtes Pergament,eine
grosse Steinschleuder,einen leeren Wasse
rkrug
450 DATA ein Paar Zauberschuhe,eine Schr
ifttafel,einen Zwergendolch,eine grosse
Kugel,ein rotes Kaestchen,einen boesen G
nom,einen eisernen Ring,einen Brief,eine
n roten Rubin
460 ge$(41)="goldenes Feenhaar":ge$(42)=
"einen feuerspeienden Drachen"
470 ge$(43)="eine verwitterte Holztafel"
:ge$(44)="einen silbernen Pokal"
480 ge(0)=-1:FOR i=9 TO 44:READ ge(i),we
(i):NEXT
490 DATA 28,0,29,0,8,0,24,0,2,0,26,0,20,
10,19,0,19,15,18,0,18,20,16,0,13,0,14,0,
0,0,0,15,15,5,8,5
500 DATA 9,0,11,0,12,0,1,0,1,0,1,0,3,7,4
,0,4,0,4,0,0,0,0,12,30,0,33,10,51,0,
40,0,51,44
505 '
510 ' VORSPANN 2
515 '
520 PEN 2:LOCATE 5,14:PRINT "Neues Spiel
beginnen.....1":LOCATE 5,16:PRINT
"Altes Spiel fortsetzen.....2":PEN 1:
LOCATE 10,24:PRINT CHR$(164);" 12/1985 b
y JR - SOFT"

```

# T R A U M L A N D

```

530 GOSUB 3390:INK 0,19:INK 1,0:INK 2,26
:INK 3,8:PEN 1
540 a$=INKEY$:IF a$="" THEN 540
550 IF a$="1" AND av=0 THEN 570 ELSE IF
a$="1" AND av=1 THEN 660
560 IF a$="2" THEN CLS:GOTO 3020 ELSE 54
0
570 PEN 3:LOCATE 1,24:PRINT SPC(39);" ";
:LOCATE 2,10:PRINT SPC(39)
580 PRINT " Mach dich auf den gefahrvoll
en Weg in      das zauberhafte Traumland
und suche  nach maerchenhaften Schaetzen
."
590 PRINT SPC(39):PRINT " Du kannst mich
mit einfachen Kommandos in einem oder z
wei Worten wie z.B.:"
600 PRINT " "+CHR$(24)+"Norden"+CHR$(24)
+" - "+CHR$(24)+"benuetze Tarnkappe"+CHR
$(24)+" - "+CHR$(24)+"Inventar"+CHR$(24)
610 PRINT SPC(32):PRINT " Um den aktue
llen Stand zu speichern,  musst Du "+CH
R$(24)+"save"+CHR$(24)+" eingeben, um ei
n Spiel einzulesen, musst Du "+CHR$(
24)+"load"+CHR$(24)+" eingeben."
620 IF av=0 THEN PRINT SPC(32):PRINT "
Druecke eine Taste und beginne !":av=1
:GOTO 640
630 PRINT SPC(32):PRINT " Druecke eine
Taste und fahre fort ! "
640 CALL &BBO3:PRINT SPC(32):PEN 1
650 IF INKEY$="" THEN 650
660 CLS:ol=0
665 '
670 ' KOMMANDOABFRAGE
675 '
680 IF ge(13)=-2 THEN ge(13)=-4
690 IF oa<>ol OR ve=3 THEN ol=oa:GOSUB 9
60
700 IF ge(13)=-2 THEN PRINT "Die Tarnkap
pe hat sich in Luft aufge- loest !":ge
(13)=0:in=in-1
710 IF ge(13)=-4 THEN ge(13)=-2
720 ze=ze+1:ko$="":PRINT "Und nun ";:INP
UT ko$
730 IF LEN(ko$)=0 THEN 720
740 v$="":n$="":ve=0:no=0
750 FOR i=1 TO LEN(ko$)
760 IF MID$(ko$,i,1)<>" " THEN v$=v$+MID
$(ko$,i,1):NEXT
770 IF LEN(v$)+1>=LEN(ko$) THEN 790
780 n$=UPPER$(RIGHT$(ko$,LEN(ko$)-i))
790 v$=UPPER$(v$):FOR i=1 TO 31:IF v$=UP
PER$(ve$(i)) THEN ve=i:GOTO 830
800 NEXT
810 IF n$="" THEN n$=v$
820 GOTO 840
830 IF v$=ko$ THEN 860
840 FOR i=1 TO 44:IF n$=UPPER$(no$(i)) T
HEN no=i:GOTO 860
850 NEXT
860 IF no<9 AND no<>0 AND (ve=0 OR ve=1)
THEN ve=1
870 IF LEN(n$)>0 AND no=0 AND ve<>16 THE
N ve=0:f1=1
880 IF f1=1 THEN f1=0:PRINT CHR$(24);n$;
CHR$(24);" kenne ich nicht.":GOTO 680

```



```
890 IF n$="" AND (ve>8 OR ve=2) AND ve<>
22 AND ve<>15 AND ve<>30 THEN f1=1
900 IF f1=1 THEN f1=0:PRINT "Bitte gib e
in "+CHR$(24)+"Objekt"+CHR$(24)+" an.":G
OTO 680
910 IF ve=0 THEN PRINT CHR$(24);v$;CHR$(
24);" kenne ich nicht.":GOTO 680
920 ON ve GOTO 1180,1250,680,1490,2950,3
020,1600,1640,1680,1750,1780,1940,2010,2
090,2140,2170,2880,2210,2260,2330,2400,1
550,2470,2520,2590,1360,2680,2720,2750,2
810,2590
930 ON ve-11 GOTO 1940,2010,2090,2140,21
70,2880,2210,2260,2330,2400
940 ON ve-21 GOTO 1550,2470,2520,2590,13
60,2680,2720,2750,2810,2590
945 '
950 ' ORTSBESTIMMUNG
955 '
960 CLS:PRINT:PRINT STRING$(40,CHR$(154)
):IF oa=25 THEN ge(40)=25:ge$(40)="eine
Tuer im Norden"
970 IF oa=30 THEN ge(40)=30:ge$(40)="ein
e Tuer im Sueden"
980 IF oa=6 THEN p1=1
990 IF oa=31 THEN p2=1
1000 IF oa=29 THEN p3=1
1010 IF oa<>1 AND ge(31)<>-2 THEN :PEN 2
:PAPER 1:PRINT:PRINT "Hilfe ! Ich vers
inke im Boden.":PRINT:PAPER 0:PEN 1:FOR
i=1 TO 2000:NEXT:GOTO 3140
1020 PEN 3:PRINT "Ich bin ";o$(oa):PRINT
:f1=0:PEN 2:PRINT "Ich sehe.":FOR i=9 TO
44:IF ge(i)<>oa THEN 1040
1030 IF f1=1 THEN PRINT ", ";ge$(i);:f1=1
:ELSE PRINT ge$(i):f1=1
1040 NEXT:f1=0:PEN 1:PRINT
1050 IF oa=31 THEN 1160
1060 IF in>1 AND oa=5 THEN 1080
1070 GOTO 1120
1080 FOR i=9 TO 44:IF (ge(i)=-1 OR ge(i)
=-2)AND i<>31 THEN ge(i)=29
1090 NEXT:in=1
1100 PAPER 2:PRINT:PRINT "Im Moor ist al
les verschwunden, was ich bei mir hatte
":PRINT
1110 PAPER 0
1120 f1=0:PRINT "Richtungen:":FOR i=0 TO
3:IF r(oa,i)=0 THEN 1150
1130 IF f1=1 THEN PRINT ", ";
1140 PRINT no$(i+5);:f1=1
1150 NEXT:f1=0:PRINT
1160 PRINT STRING$(40,CHR$(154)):f1=0:RE
TURN
1170 '
1180 IF no=0 THEN PRINT "Ich brauche ein
e Richtung.":GOTO 680
1190 IF no>8 THEN 2470
1200 IF no>4 THEN no=no-4
1210 no=no-1:IF r(oa,no)=0 THEN PRINT "I
n diese Richtung fuehrt "+CHR$(24)+"kein
"+CHR$(24)+" Weg.":GOTO 680
1220 IF (oa=30 OR (oa=25 AND no=0))AND t
u=0 THEN PRINT "Die Tuer ist versperrt."
:GOTO 680
1230 oa=r(oa,no):o1=0:GOTO 690
1240 '
```

```
1250 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
1260 IF no<>10 AND no<>14 AND no<>16 AND
no<>18 AND no<>36 AND no<>42 THEN PRINT
:PRINT a$(2):GOTO 680
1270 IF no=14 THEN PRINT:PRINT "Versuche
'schneide'.":GOTO 680
1280 IF ge(15)<>-1 AND ge(25)<>-1 AND ge
(33)<>-1 THEN PRINT:PRINT a$(1):GOTO 680
1290 IF no=10 THEN PRINT:PRINT "Die Raup
e ist kitzelig und lacht laut !":GOTO 69
0
1300 IF no=16 THEN PRINT:PRINT "Der Baer
brummt unwillig.":GOTO 690
1310 IF no=18 THEN PRINT:PRINT "Der Zwer
g wird boese und toetet mich !":GOTO 134
0
1320 IF no=36 THEN PRINT:PRINT "Der Gnom
verzaubert mich !":GOTO 1340
1330 IF no=42 THEN PRINT:PRINT "Der Drac
he verbrennt mich !"
1340 FOR i=1 TO 2000:NEXT:GOTO 3140
1350 '
1360 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
1370 IF ge(no)=-1 OR ge(no)=-2 THEN PRIN
T:PRINT "Habe ich dabei.":GOTO 680
1380 IF no=10 OR no=16 OR no=18 OR no=21
no=42 THEN PRINT:PRINT a$(0):GOTO 680
1390 IF no=29 OR no=14 OR (no=34 AND ge(
9)<>-3)THEN PRINT:PRINT "Das ist mir zu
schwer.":GOTO 680
1400 IF no=17 AND ge(16)=oa AND ge(13)<>
-2 THEN PRINT:PRINT "Der Baer";a$(3):GOT
O 690
1410 IF no=19 AND ge(18)=oa AND ge(13)<>
-2 THEN PRINT:PRINT "Der Zwerg";a$(3):GO
TO 690
1420 IF no=35 AND ge(36)=oa AND ge(13)<>
-2 THEN PRINT:PRINT "Der Gnom";a$(3):GOT
O 690
1430 IF ge(10)=oa AND ge(13)<>-2 THEN PR
INT:PRINT "Die Raupe";a$(3):GOTO 690
1440 IF no=44 AND ge(42)=oa AND ge(13)<>
-2 THEN PRINT:PRINT "Der Drache";a$(3):G
OTO 690
1450 IF no=32 OR no=43 THEN PRINT:PRINT"
Das kann ich nicht erreichen.":GOTO 680
1460 IF in+1>7 THEN PRINT:PRINT "Ich hab
e zuviel zu tragen.":PRINT "Ich muesste
etwas weglegen.":GOTO 680
1470 in=in+1:ge(no)=-1:PRINT:PRINT "Gut.
":GOTO 690
1480 '
1490 IF in=0 THEN PRINT:PRINT "Ich habe
nichts dabei.":GOTO 680
1500 PEN 1:f1=0:PRINT "Im Moment habe ic
h dabei.":
1510 FOR i=9 TO 44:IF ge(i)<>-1 AND ge(i)
<>-2 THEN 1530
1520 IF f1=1 THEN PRINT ", ";ge$(i);:f1=1
:ELSE PRINT ge$(i):f1=1
1530 NEXT:PRINT:f1=0:PEN 1:GOTO 680
1540 '
1550 PRINT:PRINT "Ich kann nur die Anlei
tung wiederholen."
```



```

1560 PRINT "Willst du das ? (j/n)"
1570 IF INKEY(45)>-1 THEN 570
1580 IF INKEY(46)>-1 THEN 680 ELSE 1570
1590 '
1600 pu=0:FOR i=9 TO 44:IF ge(i)=1 THEN
pu=pu+we(i)
1610 NEXT:PRINT:PRINT "Du hast";pu;"Punk
te von maximal 170."
1620 PRINT "Dazu gibt es einen Bonus von
bis zu 30 Punkten, wenn Du aufhoerst,
also insge- samt ein Maximum von 200 Pun
kten.":GOTO 680
1630 '
1640 a$="":PRINT CHR$(24)+"Wirklich (j/n
) ?"+CHR$(24);:INPUT"a$,a$:a$=UPPER$(a$)
1650 IF a$="J" THEN 3140
1660 IF a$="N" THEN PRINT "Na also !":GO
TO 680:ELSE GOTO 1640
1670 '
1680 f=1:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
1690 IF no<>34 OR ge$(34)="eine goldene
Kugel" THEN PRINT:PRINT "Gut. Nichts pas
siert.":in=in-1:ge(no)=oa:GOTO 690
1700 IF ge(36)<>oa THEN PRINT:PRINT "Die
Kugel faellt auf den Boden, und zer
splittert.":in=in-1:ge(34)=0:GOTO 690
1710 PRINT:PRINT "Die Kugel zerquetscht
den Gnom und"
1720 PRINT "verwandelt sich. Sag 'sie
h'.":ge$(34)="eine goldene Kugel"
1730 in=in-1:ge(36)=0:ge(34)=oa:we(34)=1
1:GOTO 690
1740 '
1750 f=0:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
1760 PRINT:PRINT "Bitte etwas genauer.":
GOTO 680
1770 '
1780 f=0:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
1790 IF no=11 OR no=12 OR no=17 OR no=21
OR no=28 OR no=32 OR no=38 OR no=43 THE
N 1810
1800 PRINT:PRINT "Da steht nichts geschr
ieben.":GOTO 680
1810 PRINT:PEN 2:PRINT "Die Schrift besa
gt.":PEN 3
1820 IF no=11 THEN PRINT "Messer sind zu
m stechen da !"
1830 IF no=12 THEN PRINT "Drachen lassen
sich nicht verjagen."
1840 IF no=12 THEN PRINT "Sie sind unver
wundbar und unsterblich."
1850 IF no=17 THEN PRINT "Dinge drehen s
ich um Ringe."
1860 IF no=21 THEN PRINT "Ein Zauberer w
ar hier.":PRINT "Simsalabim..."
1870 IF no=28 THEN PRINT "Voegel wurden
vor langer Zeit in Zwerge verwandelt."
1880 IF no=32 THEN PRINT "Fremdling, der
du betrittst dies Land ! Suche nach den
verlorenen "+CHR$(24)+"Schaetzen"+CHR$(
24)+" !"
1890 IF no=32 THEN PRINT "Bringe sie hie
rher und leg sie hin. Sag 'traumland
' !"

```

# T R A U M L A N D

```

1900 IF no=38 THEN PRINT "Pflanzen brauc
hen Wasser !"
1910 IF no=43 THEN PRINT "Huete dich vor
dem Drachen !"
1920 f1=0:PEN 1:GOTO 680
1930 '
1940 f=1:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
1950 IF no<>30 AND no<>35 AND no<>44 THE
N PRINT:PRINT a$(0):GOTO 680
1960 IF no=35 OR no=44 THEN PRINT:PRINT
"Das ist dafuer nicht geeignet.":GOTO 68
0
1970 IF oa<>3 AND oa<>17 AND oa<>35 THEN
PRINT:PRINT "Wo ?????":GOTO 680
1980 IF oa=30 OR oa=35 THEN PRINT:PRINT
"ich kann das Wasser nicht erreichen.":G
OTO 680
1990 PRINT:PRINT "Gut.":ge$(30)="einen v
ollen Wasserkrug":GOTO 690
2000 '
2010 f=0:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
2020 IF no=40 THEN PRINT:PRINT "Versuche
'sperre'.":GOTO 680
2030 f=1:GOSUB 3110:IF f=1 THEN f=0:GO
TO 680
2040 IF no<>35 THEN PRINT:PRINT a$(2):GO
TO 680
2050 IF ge(38)<>0 THEN PRINT:PRINT "Gut.
Es ist leer.":GOTO 690
2060 IF ge(26)<>-1 THEN PRINT:PRINT a$(1
):GOTO 680
2070 PRINT "Zwei Dinge fallen heraus. Sa
g 'sieh'.":ge(38)=oa:ge(39)=oa:GOTO 690
2080 '
2090 f=1:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
2100 IF no<>13 AND no<>31 THEN PRINT:PRI
NT "Bitte etwas genauer.":GOTO 680
2110 IF no=13 THEN PRINT:PRINT "Gut. Ich
setze die Tarnkappe auf.":ge(13)=-2:GOT
O 680
2120 IF no=31 THEN PRINT:PRINT "Gut. Ich
ziehe die Zauberschuhe an.":ge(31)=-2:G
OTO 690
2130 '
2140 IF ge(18)<>oa THEN PRINT:PRINT "Nic
hts passiert.":GOTO 680
2150 PRINT:PRINT "Der Zwerg verwandelt s
ich in einen bunten Vogel und flieg
t davon.":ge(18)=0:GOTO 690
2160 '
2170 PRINT:PRINT "Gut,...":n$
2180 IF n$=ve$(15) THEN 2140
2190 PRINT "Nichts passiert.":GOTO 680
2200 '
2210 f=1:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680
2220 IF no=34 THEN PRINT:PRINT "Versuche
'wirf'.":GOTO 680
2230 IF no=20 THEN PRINT:PRINT "Versuche
'schleudere'.":GOTO 680
2240 PRINT a$(0):GOTO 680
2250 '
2260 f=0:GOSUB 3090:IF f=1 THEN f=0:GO
TO 680

```





```

2270 IF no<>10 AND no<>16 AND no<>18 AND
no<>36 AND no<>42 THEN PRINT:PRINT a$(0)
):GOTO 680
2280 IF no<>16 THEN PRINT:PRINT "Hat kei
nen Hunger.":GOTO 680
2290 IF ge(23)<>-1 THEN PRINT:PRINT "Der
Baer ist sehr hungrig und frisst mic
h !":FOR i=1 TO 2000:NEXT:GOTO 3140
2300 PRINT:PRINT "Der Baer verschlingt d
ie Beeren und laeuft davon."
2310 in=in-1:ge(16)=0:ge(23)=0:GOTO 690
2320 '
2330 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2340 IF no<>14 THEN PRINT:PRINT a$(2):GO
TO 680
2350 IF ge(33)<>-1 THEN PRINT:PRINT a$(1)
):GOTO 680
2360 IF ge(37)<>0 THEN 2380
2370 PRINT:PRINT "Ich habe die Frucht ge
teilt und etwas faellt heraus. Sag 'si
eh'.":ge(37)=oa:GOTO 690
2380 PRINT:PRINT "Die Frucht ist bereits
geteilt.":GOTO 680

```

```

2610 IF ve=31 AND no=30 AND ge$(30)="ein
en vollen Wasserkrug" THEN f1=2
2620 IF f1=2 THEN f1=1:ge$(30)="einen le
eren Wasserkrug"
2630 IF f1=1 THEN f1=0:PRINT:PRINT "Aaaa
h....Gut !":GOTO 690
2640 IF ve=31 AND no=30 THEN PRINT:PRINT
"Der Krug ist leer.":GOTO 680
2650 IF ve=25 AND no=9 THEN ge(9)=-3:PRI
NT:PRINT "Brrrr....Schmeckt ja scheussli
ch !":in=in-1:PRINT "Ich fuehle mich jet
zt besonders stark !":GOTO 690
2660 PRINT:PRINT a$(0):GOTO 680
2670 '
2680 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2690 IF no<>10 AND no<>16 AND no<>18 AND
no<>36 AND no<>42 THEN PRINT:PRINT a$(0)
):GOTO 680
2700 PRINT "Laesst sich nicht verjagen."
:GOTO 680
2710 '
2720 f=1:GOSUB 3110:IF f1=1 THEN f1=0:GO
TO 680

```

**Nur noch wenige Zeilen zum Eintippen und Sie können sich ins „Schneider-Traumland“ begeben.**

```

2390 '
2400 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2410 IF no<>40 AND no<>35 THEN PRINT:PRI
NT a$(2):GOTO 680
2420 IF no=35 THEN PRINT:PRINT "Versuche
'oeffne'.":GOTO 680
2430 IF tu=1 THEN PRINT:PRINT "Ist schon
offen !":GOTO 680
2440 IF ge(26)<>-1 THEN PRINT:PRINT "Ich
habe keinen Schluessel.":GOTO 680
2450 PRINT:PRINT "Gut.":tu=1:GOTO 690
2460 '
2470 IF oa=31 AND no=27 THEN oa=9:GOTO 6
90
2480 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2490 IF no<>27 THEN PRINT:PRINT a$(0):GO
TO 680
2500 oa=31:GOTO 690
2510 '
2520 f=1:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2530 IF no<>20 AND no<>34 THEN PRINT:PRI
NT a$(2):GOTO 680
2540 IF no=34 THEN PRINT:PRINT "Versuche
'wirf'.":GOTO 680
2550 IF ge(29)<>oa THEN PRINT:PRINT "Wom
it ??":GOTO 680
2560 PRINT:PRINT "Gut.":in=in-1:ge(20)=2
9:ge(10)=0
2570 GOTO 690
2580 '
2590 f=1:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2600 IF ve=25 AND no=23 THEN f1=1:ge(23)
=0:in=in-1

```

```

2730 in=in-1:PRINT:PRINT "Gut.":ge(no)=o
a:GOTO 680
2740 '
2750 f=1:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2760 IF no<>37 THEN PRINT:PRINT "Gut, ni
chts passiert.":GOTO 690
2770 PRINT:PRINT "Alles beginnt sich zu
drehen !":PRINT "Der Ring faellt mir aus
der Hand !"
2780 PRINT "Sag 'sieh'.":ge(37)=oa:we(37
)=16
2790 ge$(37)="einen Ring aus Goldfaeden"
:in=in-1:GOTO 690
2800 '
2810 IF oa=31 AND (no=0 OR no=27) THEN 28
40
2820 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2830 PRINT:PRINT a$(0):GOTO 680
2840 IF INT(4*RND+1)=2 THEN 2860
2850 oa=9:GOTO 690
2860 PRINT:PRINT "Plumps..AUA ! Ich bin
heruntergefallen.":FOR i=1 TO 2000:NEXT:
GOTO 3140
2870 '
2880 f=0:GOSUB 3090:IF f1=1 THEN f1=0:GO
TO 680
2890 IF no<>22 AND no<>14 AND no<>27 THE
N PRINT:PRINT a$(0):GOTO 680
2900 IF ge(30)<>-1 OR ge$(30)="einen lee
ren Wasserkrug" THEN PRINT:PRINT "Ich ha
be nichts zu giessen.":GOTO 680
2910 IF no=22 THEN PRINT:PRINT "Der Stra
uch ginnt zu bluehen und es wachsen
rote Beeren.":PRINT "Dann verwandelt er
sich. Sag 'sieh'.":ge(22)=0:ge(23)=oa:ge

```



```
(24)=oa:GOTO 2930
2920 PRINT:PRINT "Gut."
2930 ge$(30)="einen leeren Wasserkrug":G
OTO 690
2940 '
2950 SPEED WRITE 1:PRINT:PRINT "Lege ein
e Kasette zum abspeichern ein.":PRINT:O
PENOUT "traumdat"
2960 FOR i=9 TO 44:PRINT #9,ge(i):NEXT
2970 PRINT #9,oa:PRINT #9,tu:PRINT #9,in
:PRINT #9,p1:PRINT #9,p2:PRINT #9,p3
2980 PRINT #9,ze:PRINT #9,we(34):PRINT #
9,we(37):PRINT #9,ge$(30)
2990 PRINT #9,ge$(34):PRINT #9,ge$(37)
3000 CLOSEOUT:ol=0:GOTO 680
3010 '
3020 PRINT:PRINT "Lege eine Kasette mit
einem Spiel ein.":PRINT:OPENIN "traumda
t"
3030 FOR i=9 TO 44:INPUT #9,ge(i):NEXT
3040 INPUT #9,oa:INPUT #9,tu:INPUT #9,in
:INPUT #9,p1:INPUT #9,p2:INPUT #9,p3
3050 INPUT #9,ze:INPUT #9,we(34):INPUT #
9,we(37):INPUT #9,ge$(30)
3060 INPUT #9,ge$(34):INPUT #9,ge$(37)
3070 CLOSEIN:ol=0:GOTO 680
3080 '
3090 IF ge(no)<>oa AND ge(no)<>-1 AND ge
(no)<>-2 THEN fl=1
3100 IF fl=1 THEN PRINT:PRINT "So etwas
sehe ich hier nicht.":RETURN
3110 IF f=1 AND ge(no)<>-1 AND ge(no)<>-
2 THEN PRINT:PRINT "So etwas habe ich ni
cht bei mir.":fl=1
3120 RETURN
3130 '
3140 pu=0:FOR i=9 TO 44:IF ge(i)=1 THEN
pu=pu+we(i)
3150 NEXT:IF pu<>170 THEN 3190
3160 IF ze<500 THEN pu=pu+7
3170 IF ze<400 THEN pu=pu+7
3180 IF ze<300 THEN pu=pu+7
3190 IF p1=1 THEN pu=pu+2
3200 IF p2=1 THEN pu=pu+3
3210 IF p3=1 THEN pu=pu+4
3220 CLS:BORDER 26,0:LOCATE 5,4:PEN 1:PA
PER 2:PRINT "Du hast";pu;"von 200 Punkte
n !"
3230 PAPER 3:PEN 2:LOCATE 14,9:PRINT "-
Rang ";
3240 IF pu=0 THEN PRINT "10 -":GOTO 3340
3250 IF pu<25 THEN PRINT "9 -":GOTO 3340

3260 IF pu<50 THEN PRINT "8 -":GOTO 3340

3270 IF pu<75 THEN PRINT "7 -":GOTO 3340

3280 IF pu<100 THEN PRINT "6 -":GOTO 334
0
3290 IF pu<125 THEN PRINT "5 -":GOTO 334
0
3300 IF pu<150 THEN PRINT "4 -":GOTO 334
0
```

```
3310 IF pu<175 THEN PRINT "3 -":GOTO 334
0
3320 IF pu<200 THEN PRINT "2 -":GOTO 334
0
3330 IF pu=200 THEN PRINT "1 -"
3340 LOCATE 8,16:PAPER 0:PEN 1:PRINT "No
ch ein Spiel (j/n) ?"
3350 a$=UPPER$(INKEY$):IF a$="" THEN 335
0
3360 IF a$="J" THEN RUN
3370 IF a$<>"N" THEN 3350 ELSE CALL &BCO
2:PAPER 0:PEN 1:MODE 2:END
3380 '
3390 '
3400 GOSUB 3420:a=REMAIN (0):PEN 1:PAPER
0:RETURN
3410 RETURN
3420 FOR uxi=1 TO 2:FOR ixi=1 TO 2:FOR j
xi=1 TO 2:RESTORE 3490:GOSUB 3550:NEXT
3430 FOR jxi=1 TO 2:RESTORE 3500:GOSUB 3
550:NEXT:NEXT
3440 FOR ixi=1 TO 2:RESTORE 3510:GOSUB 3
550:NEXT
3450 FOR ixi=1 TO 2:RESTORE 3520:GOSUB 3
550:NEXT
3460 FOR ixi=1 TO 2:RESTORE 3530:GOSUB 3
550:NEXT
3470 FOR ixi=1 TO 2:RESTORE 3540:GOSUB 3
550:NEXT:NEXT
3480 FOR ixi=7 TO 0 STEP -0.1:SOUND 7,50
-ixi,6,ixi,,,3:NEXT:RETURN
3490 DATA 123,247,494,123,247,494,740,74
3,737,494,496,492,659,661,657,494,496,49
2,554,556,552,587,589,585
3500 DATA 110,220,440,110,220,440,740,74
2,738,440,442,438,659,661,657,440,442,43
8,587,589,585,659,661,657
3510 DATA 147,294,587,147,294,587,880,88
2,878,587,589,585,740,742,738,587,589,58
5,659,661,657,740,742,738
3520 DATA 123,247,590,123,247,590,988,99
0,986,587,589,585,740,742,738,587,589,58
5,659,661,657,740,742,738
3530 DATA 196,392,587,196,392,587,988,99
0,986,587,589,585,784,786,782,587,589,58
5,740,742,738,784,786,782
3540 DATA 139,277,554,139,277,554,659,66
1,657,740,742,738,932,934,930,740,742,73
8,659,661,657,554,556,552
3550 FOR kxi=1 TO 8:READ no1,no2,no3:IF
kxi=1 OR kxi=2 THEN 3570 ELSE SOUND 49,1
25000/no1,17,15:SOUND 42,125000/no2,17,1
5:SOUND 28,125000/no3,17,15
3560 NEXT:RETURN
3570 SOUND 49,125000/no1,14,15:SOUND 42,
125000/no2,14,15:SOUND 28,125000/no3,14,
15:GOTO 3560
```

Listing: Ende

**Sie werden feststellen, daß sich die Mühen des Eintippens gelohnt haben**



## CHECKSUMMEN ZUM SCHNEIDER-PROGRAMM „TRAUMLAND“

## ZEILENNR.: SUMMEN

10- 20: 38660	630- 640: 41045	1350- 1360: 55718	2110- 2120: 17500	2870- 2880: 55718
30- 40: 73473	650- 660: 11493	1370- 1380: 61904	2130- 2140: 30294	2890- 2900: 77672
50- 60: 25534	665- 670: 12574	1390- 1400: 28846	2150- 2160: 2980	2910- 2920: 5767
70- 75: 2270	675- 680: 38150	1410- 1420: 14072	2170- 2180: 26650	2930- 2940: 7835
80- 85: 2483	690- 700: 27028	1430- 1440: 33693	2190- 2200: 27585	2950- 2960: 96510
90- 100: 35191	710- 720: 44332	1450- 1460: 73637	2210- 2220: 120857	2970- 2980: 74074
110- 120: 114472	730- 740: 56429	1470- 1480: 19810	2230- 2240: 31664	2990- 3000: 31893
130- 140: 74720	750- 760: 50764	1490- 1500: 21207	2250- 2260: 55718	3010- 3020: 25191
150- 160: 55350	770- 780: 75987	1510- 1520: 64426	2270- 2280: 19621	3030- 3040: 86799
170- 180: 58145	790- 800: 41759	1530- 1540: 16065	2290- 2300: 57248	3050- 3060: 36633
190- 195: 61967	810- 820: 20117	1550- 1560: 49081	2310- 2320: 64536	3070- 3080: 9460
200- 205: 7209	830- 840: 58254	1570- 1580: 28758	2330- 2340: 85700	3090- 3100: 102016
210- 220: 21887	850- 860: 49569	1590- 1600: 29246	2350- 2360: 53215	3110- 3120: 34377
230- 240: 99709	870- 880: 72652	1610- 1620: 36858	2370- 2380: 13577	3130- 3140: 29246
250- 260: 58022	890- 900: 65455	1630- 1640: 29629	2390- 2400: 55718	3150- 3160: 52400
265- 270: 3046	910- 920: 72816	1650- 1660: 61067	2410- 2420: 67450	3170- 3180: 72692
275- 280: 30690	930- 940: 14772	1670- 1680: 55724	2430- 2440: 129281	3190- 3200: 63578
290- 300: 98741	945- 950: 12169	1690- 1700: 65145	2450- 2460: 22429	3210- 3220: 89572
305- 310: 3667	955- 960: 32322	1710- 1720: 44781	2470- 2480: 111273	3230- 3240: 38264
315- 320: 30471	970- 980: 79926	1730- 1740: 57870	2490- 2500: 38201	3250- 3260: 37095
330- 340: 103404	990- 1000: 35676	1750- 1760: 96419	2510- 2520: 55724	3270- 3280: 38027
350- 355: 32420	1010- 1020: 66424	1770- 1780: 55718	2530- 2540: 125455	3290- 3300: 38959
360- 365: 6438	1030- 1040: 16664	1790- 1800: 17888	2550- 2560: 78186	3310- 3320: 39891
370- 380: 65160	1050- 1060: 35019	1810- 1820: 59264	2570- 2580: 1357	3330- 3340: 71136
390- 400: 100329	1070- 1080: 29835	1830- 1840: 65747	2590- 2600: 58765	3350- 3360: 36892
405- 410: 9063	1090- 1100: 65626	1850- 1860: 18133	2610- 2620: 21276	3370- 3380: 44144
415- 420: 30339	1110- 1120: 13083	1870- 1880: 56315	2630- 2640: 58915	3390- 3400: 24860
430- 440: 65242	1130- 1140: 36731	1890- 1900: 18513	2650- 2660: 28187	3410- 3420: 64926
450- 460: 46140	1150- 1160: 33178	1910- 1920: 25842	2670- 2680: 55718	3430- 3440: 67979
470- 480: 64460	1170- 1180: 29208	1930- 1940: 55724	2690- 2700: 67403	3450- 3460: 63424
490- 500: 99375	1190- 1200: 42041	1950- 1960: 84701	2710- 2720: 55944	3470- 3480: 95711
505- 510: 6095	1210- 1220: 50430	1970- 1980: 61348	2730- 2740: 26480	3490- 3500: 91604
515- 520: 34111	1230- 1240: 52015	1990- 2000: 49515	2750- 2760: 77922	3510- 3520: 94259
530- 540: 44989	1250- 1260: 70440	2010- 2020: 63606	2770- 2780: 21498	3530- 3540: 93738
550- 560: 74422	1270- 1280: 39687	2030- 2040: 86136	2790- 2800: 7037	3550- 3560: 64584
570- 580: 85099	1290- 1300: 70773	2050- 2060: 50822	2810- 2820: 112241	3570- : 64790
590- 600: 114324	1310- 1320: 51361	2070- 2080: 26875	2830- 2840: 26245	
610- 620: 62085	1330- 1340: 40000	2090- 2100: 119893	2850- 2860: 54936	

Gesamtchecksumme: 5066

## EIN 3-D SPIEL FÜR CPC 464/664/6128

## TRITON

Triton ist ein Spiel der Sonderklasse. 20! (Zwanzig!) Sprites sorgen für eine tolle Graphik und prima Animation.

Der irdische Geheimdienst meldete, daß vor wenigen Tagen eine Rebellion auf dem Planeten Triton stattgefunden hat. Sie bekommen den Auftrag, dort mit Ihrem Raumschiff nach dem Rechten zu sehen. Doch kaum sind Sie gelandet, werden Sie auch schon angegriffen. Jetzt heißt es schnell und sicher reagieren – schießen Sie die ferngesteuerten Angreifer ab, oder deren Bomben. Denn nach 3 Treffern ist Ihr Raumschiff zerstört. Die Steuerung erfolgt durch den Joystick.

WICHTIG: Speichern Sie das Programm vor dem ersten Start ab, da es sich bei einem Eingabefehler eventuell selbst zerstört.



„Ich hätte nie geglaubt, daß ich dieser Spielaufgabe gewachsen sein könnte und dann habe ich es doch geschafft!“





Das Spiel besteht durch die hohe Geschwindigkeit und die gute Animation der bewegten Objekte.

```
1 *****
2 *      T R I T O N      *
3 *      (c)  1986      *
4 *    OTTFRIED SCHMIDT  *
5 *****
```

```
6
7
10 DEFINT a-z:DEFREAL s:DEFREAL h:hsc=30
```

```
O:nam$=" "
```

```
20 MODE 0:SYMBOL AFTER 256:MEMORY &918F:
SYMBOL AFTER 199
```

```
30 INK 0,0:INK 1,6:INK 2,11:INK 3,24:INK
4,23
```

```
40 INK 5,6:INK 6,4:INK 7,2,1:INK 8,1,2:I
NK 9,15
```

```
50 INK 10,6:INK 11,15:INK 12,8:INK 13,24
```

```
60 RESTORE 1950:GOSUB 1940
```

```
70 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
80 DATA 0,0,0,0,0,0,1,0,0,0,0
```

```
90 DATA 0,0,0,2,2,2,2,2,0,0,0
```

```
100 DATA 0,0,0,0,3,3,3,0,0,0,0
```

```
110 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
120 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
130 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
140 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
150 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
160 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
170 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
180 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
190 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
200
```

```
210 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
220 DATA 0,0,0,0,1,1,1,0,0,0,0
```

```
230 DATA 0,0,2,2,2,2,2,2,2,0,0
```

```
240 DATA 0,2,2,2,2,2,2,2,2,2,2
```

```
250 DATA 0,0,0,3,3,3,3,3,0,0,0
```

```
260 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
270 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
280 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
290 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
300 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
310 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
320 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
330 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
340
```

```
350 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
360 DATA 0,0,2,2,1,1,0,0,0,0,0
```

```
370 DATA 0,2,2,2,2,2,1,2,0,0,0
```

```
380 DATA 0,0,0,3,2,2,2,2,2,0,0
```

```
390 DATA 0,0,0,0,3,3,3,2,2,2,2
```

```
400 DATA 0,0,0,0,0,0,0,3,0,2,0
```

```
410 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
420 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
430 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
440 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
450 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
460 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
470 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
480
```

```
490 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
500 DATA 0,2,2,0,0,0,0,0,0,0,0
```

```
510 DATA 0,2,2,2,1,0,0,0,0,0,0
```

```
520 DATA 0,0,2,2,2,1,1,0,0,0,0
```

```
530 DATA 0,0,0,3,2,2,2,1,0,0,0
```

```
540 DATA 0,0,0,0,3,3,2,2,2,0,0
```

```
550 DATA 0,0,0,0,0,3,3,2,2,0,0
```

```
560 DATA 0,0,0,0,0,0,0,3,2,2,0
```

```
570 DATA 0,0,0,0,0,0,0,0,2,2,0
```

```
580 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
590 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
600 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
610 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
620
```

```
630 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
640 DATA 0,2,2,0,0,0,0,0,0,0,0
```

```
650 DATA 0,2,2,2,1,0,0,0,0,0,0
```

```
660 DATA 0,0,2,2,2,1,0,0,0,0,0
```

```
670 DATA 0,0,0,3,2,2,1,0,0,0,0
```

```
680 DATA 0,0,0,0,3,2,2,1,0,0,0
```

```
690 DATA 0,0,0,0,0,3,2,2,0,0,0
```

```
700 DATA 0,0,0,0,0,0,3,2,2,0,0
```

```
710 DATA 0,0,0,0,0,0,0,2,2,2,0
```

```
720 DATA 0,0,0,0,0,0,0,0,2,2,0
```

```
730 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
740 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
750 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
760
```

```
770 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
780 DATA 0,0,2,2,0,0,0,0,0,0,0
```

```
790 DATA 0,2,2,2,2,1,0,0,0,0,0
```

```
800 DATA 0,0,2,2,2,2,1,0,0,0,0
```

```
810 DATA 0,0,2,3,2,2,2,1,0,0,0
```

```
820 DATA 0,0,0,2,3,2,2,2,1,0,0
```

```
830 DATA 0,0,0,0,2,3,2,2,2,0,0
```

```
840 DATA 0,0,0,0,0,2,3,2,2,0,0
```

```
850 DATA 0,0,0,0,0,0,2,2,2,2,0
```

```
860 DATA 0,0,0,0,0,0,0,2,2,2,0
```

```
870 DATA 0,0,0,0,0,0,0,0,2,0,0
```

```
880 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
890 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
900
```

```
910 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
920 DATA 0,0,0,2,0,0,0,0,0,0,0
```

```
930 DATA 0,0,2,2,2,2,0,0,0,0,0
```

```
940 DATA 0,2,2,2,2,2,0,0,0,0,0
```

```
950 DATA 0,2,2,2,2,1,2,2,0,0,0
```

```
960 DATA 0,2,2,3,2,2,2,2,0,0,0
```

```
970 DATA 0,0,2,3,3,2,2,1,2,0,0
```

```
980 DATA 0,0,2,2,3,3,2,2,2,2,0
```

```
990 DATA 0,0,0,2,2,3,3,2,2,2,0
```

```
1000 DATA 0,0,0,0,2,2,3,3,2,2,0
```

```
1010 DATA 0,0,0,0,0,2,2,2,2,2,0
```

```
1020 DATA 0,0,0,0,0,0,2,2,2,0,0
```

```
1030 DATA 0,0,0,0,0,0,0,2,0,0,0
```

```
1040
```

```
1050 DATA 0,0,0,2,2,2,0,0,0,0,0
```

```
1060 DATA 0,0,2,2,2,2,2,0,0,0,0
```

```
1070 DATA 0,2,2,2,2,1,2,2,0,0,0
```

```
1080 DATA 0,2,2,2,2,1,2,2,2,0,0
```

```
1090 DATA 0,2,2,3,2,2,2,1,2,0,0
```

```
1100 DATA 0,2,3,3,3,2,2,1,2,2,0
```

```
1110 DATA 0,2,3,3,3,1,2,2,2,2,0
```

```
1120 DATA 0,0,2,2,1,1,3,2,2,2,0
```

```
1130 DATA 0,0,2,2,2,1,3,3,2,2,0
```

```
1140 DATA 0,0,0,2,2,2,3,3,2,2,0
```

```
1150 DATA 0,0,0,0,2,2,3,2,2,0,0
```

```
1160 DATA 0,0,0,0,0,2,2,3,2,0,0
```

```
1170 DATA 0,0,0,0,0,0,2,2,2,0,0
```

```
1180
```

```
1190 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1200 DATA 0,0,0,0,7,0,0,0,0,0,0
```

```
1210 DATA 0,0,0,0,8,0,0,0,0,0,0
```

```
1220 DATA 0,0,0,0,7,0,0,0,0,0,0
```

```
1230 DATA 0,0,0,0,8,0,0,0,0,0,0
```

```
1240 DATA 0,0,0,5,7,5,0,0,0,0,0
```

```
1250 DATA 7,8,7,0,0,0,7,8,7,0,0
```

```
1260 DATA 0,0,0,5,7,5,0,0,0,0,0
```

```
1270 DATA 0,0,0,0,8,0,0,0,0,0,0
```

```
1280 DATA 0,0,0,0,7,0,0,0,0,0,0
```

```
1290 DATA 0,0,0,0,8,0,0,0,0,0,0
```

```
1300 DATA 0,0,0,0,7,0,0,0,0,0,0
```

```
1310 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1320
```

```
1330 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1340 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1350 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1360 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1370 DATA 0,0,0,0,6,0,6,0,0,0,0
```

```
1380 DATA 0,0,0,0,6,6,6,0,0,0,0
```

```
1390 DATA 0,0,0,0,6,0,0,0,0,0,0
```

```
1400 DATA 0,0,0,0,6,0,0,0,0,0,0
```

```
1410 DATA 0,0,0,0,6,7,6,0,0,0,0
```

```
1420 DATA 0,0,0,0,6,7,6,0,0,0,0
```

```
1430 DATA 0,0,0,0,6,0,0,0,0,0,0
```

```
1440 DATA 0,0,0,0,6,0,0,0,0,0,0
```

```
1450 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1460
```

```
1470
```

```
1480 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1490 DATA 0,0,0,0,10,0,0,0,0,0,0
```

```
1500 DATA 0,0,0,0,0,0,0,10,0,0,0
```

```
1510 DATA 0,0,0,10,0,0,12,0,0,0,0
```

```
1520 DATA 0,10,0,0,0,0,0,0,0,0,0
```

```
1530 DATA 0,0,12,0,11,11,0,0,0,0,0
```

```
1540 DATA 0,0,0,0,13,11,0,0,0,0,0
```

```
1550 DATA 0,0,0,0,11,11,0,0,10,0,0
```

```
1560 DATA 0,0,0,13,0,13,0,0,0,0,0
```

```
1570 DATA 0,0,0,0,0,0,0,10,0,0,0
```

```
1580 DATA 0,10,0,0,0,0,0,0,0,0,0
```

```
1590 DATA 0,0,0,0,12,0,0,0,0,0,0
```

```
1600 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1610
```

```
1620 DATA 0,0,0,0,0,10,0,0,0,12,0
```

```
1630 DATA 0,10,0,0,0,0,0,0,10,0,0
```

```
1640 DATA 0,0,12,0,0,11,0,11,0,0,0
```

```
1650 DATA 0,0,0,0,0,0,0,0,0,0,0
```

```
1660 DATA 0,10,0,0,0,0,0,0,0,0,0
```

```
1670 DATA 0,0,0,0,0,13,0,0,12,10,0
```

```
1680 DATA 0,11,0,0,13,0,0,0,0,0,0
```

```
1690 DATA 10,0,0,0,0,0,0,11,0,0,0
```

```
1700 DATA 0,0,0,11,0,0,0,0,0,0,0
```

```
1710 DATA 0,0,12,0,0,0,0,11,0,0,0
```

```
1720 DATA 0,0,0,0,0,0,0,0,12,0,0
```

```
1730 DATA 0,0,12,0,0,0,10,0,0,0,0
```



```
1740 DATA 0,10,0,0,0,0,10,0,0,0
1750 x=399:y=1
1760 FOR i=1 TO 10*13:FOR l=1 TO 10
1770 READ a:PLOT y,x,a
1780 y=y+4:NEXT l
1790 x=x-2:y=1:NEXT i
1800 CALL &9200
1810 RESTORE
1820 x=399:y=39
1830 FOR i=1 TO 8*13:FOR l=1 TO 10
1840 READ a:PLOT y,x,a
1850 y=y-4:NEXT l
1860 x=x-2:y=39:NEXT i
1870 RESTORE 1480:FOR i=1 TO 2*13:FOR l=
1 TO 10
1880 READ a:PLOT y,x,a
1890 y=y-4:NEXT l
1900 x=x-2:y=39:NEXT i
1910 CALL &9200
1920 LOCATE 1,20
1930 GOTO 2030
1940 GOTO 1990
1950 DATA 21,00,C0,ED,4B,FE,9F,E5,16,82
1960 DATA 1E,05,7E,02,03,23,1D,20,F9,E1
1970 DATA CD,26,BC,E5,15,20,EF,E1,ED,43
1980 DATA FE,9F,00,C9
1990 FOR i=&9200 TO &9221:READ a$
2000 POKE i,VAL("&" + a$):NEXT i
2010 POKE &9FFE,0:POKE &9FFF,&A0
2020 RESTORE:RETURN
2030
2040 GOTO 2930
2050 DATA 3A,90,90,32,92,90,3A,91,90,32
2060 DATA 93,90,CD,24,BB,CB,67,CA,3D,95
2070 DATA 67,CB,4C,CC,4F,92,CB,44,CC,5B
2080 DATA 92,CB,5C,CC,43,92,CB,54,CC,37
2090 DATA 92,CD,67,92,CD,24,BB,CB,67,C2
2100 DATA F2,94,C3,4C,95,3A,90,90,FE,84
2110 DATA CB,3C,3C,32,90,90,C9,3A,90,90
2120 DATA FE,16,CB,3D,3D,32,90,90,C9,3A
2130 DATA 91,90,FE,B4,CB,3C,3C,32,91,90
2140 DATA C9,3A,91,90,FE,6E,CB,3D,3D,32
2150 DATA 91,90,C9,16,00,26,00,3A,92,90
2160 DATA 5F,3A,93,90,6F,2D,1D,CD,1D,BC
2170 DATA CD,90,92,16,00,26,00,3A,90,90
2180 DATA 5F,3A,91,90,6F,2D,1D,CD,1D,BC
2190 DATA CD,90,92,C9,01,08,A2,E5,16,0D
2200 DATA 1E,05,0A,AE,77,03,23,1D,20,FB
2210 DATA E1,CD,26,BC,E5,15,20,EE,E1,C9
2220 DATA 3E,00,32,9B,90,3A,94,90,FE,01
2230 DATA CA,66,94,2A,97,90,22,99,90,2A
2240 DATA A0,90,22,A2,90,3A,9E,90,FE,00
2250 DATA 2B,1E,FE,01,2B,0E,3A,97,90,FE
2260 DATA 86,CA,3D,94,3C,32,97,90,18,0C
2270 DATA 3A,97,90,FE,14,CA,3D,94,3D,32
2280 DATA 97,90,3A,9F,90,FE,00,2B,16,3A
2290 DATA 9B,90,FE,B6,20,09,2A,99,90,22
2300 DATA 97,90,C3,3D,94,3C,32,9B,90,18
2310 DATA 07,3A,9B,90,3D,32,9B,90,3A,9D
2320 DATA 90,FE,06,2B,07,3C,32,9D,90,C3
2330 DATA F3,93,AF,32,9D,90,3A,9D,90,FE
2340 DATA 07,CA,F3,93,3C,32,9C,90,2A,A0
2350 DATA 90,01,41,00,09,22,A0,90,FE,01
2360 DATA 20,08,3E,00,32,9F,90,C3,F3,93
2370 DATA FE,02,20,08,3E,00,32,9F,90,C3
2380 DATA F3,93,FE,03,20,08,3E,00,32,9F
2390 DATA 90,C3,F3,93,FE,04,20,1C,3E,01
2400 DATA 32,9F,90,3A,95,90,FE,01,28,0B
```

```
2410 DATA 3E,02,32,9E,90,C3,F3,93,3E,01
2420 DATA 32,9E,90,C3,F3,93,FE,05,20,1C
2430 DATA 3E,01,32,9F,90,3A,95,90,FE,01
2440 DATA 28,08,3E,02,32,9E,90,C3,F3,93
2450 DATA 3E,01,32,9E,90,C3,F3,93,FE,06
2460 DATA 20,1C,3E,01,32,9F,90,3A,95,90
2470 DATA FE,01,28,08,3E,02,32,9E,90,C3
2480 DATA F3,93,3E,01,32,9E,90,C3,F3,93
2490 DATA FE,07,20,1C,3E,01,32,9F,90,3A
2500 DATA 95,90,FE,01,28,08,3E,02,32,9E
2510 DATA 90,C3,F3,93,3E,01,32,9E,90,C3
2520 DATA F3,93,3E,01,3E,01,32,9F,90,3A
2530 DATA 95,90,FE,01,28,08,3E,02,32,9E
2540 DATA 90,C3,F3,93,3E,01,32,9E,90,16
2550 DATA 00,26,00,3A,99,90,5F,3A,9A,90
2560 DATA 6F,2D,1D,CD,1D,BC,ED,4B,A2,90
2570 DATA CD,26,94,26,00,16,00,3A,97,90
2580 DATA 5F,3A,98,90,6F,2D,1D,CD,1D,BC
2590 DATA ED,4B,A0,90,CD,26,94,C3,00,92
2600 DATA E5,16,0D,1E,05,0A,AE,77,03,23
2610 DATA 1D,20,FB,E1,CD,26,BC,E5,15,20
2620 DATA EE,E1,C9,3E,01,32,94,90,2A,97
2630 DATA 90,22,A6,90,2A,A0,90,22,A4,90
2640 DATA CD,C6,94,2A,97,90,22,AB,90,22
2650 DATA A6,90,21,49,A2,22,A4,90,CD,C6
2660 DATA 94,C3,00,92,2A,AB,90,22,AA,90
2670 DATA 3A,99,90,FE,50,2B,06,3D,32,A9
2680 DATA 90,18,06,3E,01,32,96,90,C9,16
2690 DATA 00,26,00,16,00,3A,AA,90,5F,3A
2700 DATA AB,90,6F,2D,1D,CD,1D,BC,CD,AC
2710 DATA 94,26,00,16,00,3A,AB,90,5F,3A
2720 DATA A9,90,6F,2D,1D,CD,1D,BC,CD,AC
2730 DATA 94,C3,00,92,01,49,A2,E5,16,0D
2740 DATA 1E,05,0A,AE,77,03,23,1D,20,FB
2750 DATA E1,CD,26,BC,E5,15,20,EE,E1,C9
2760 DATA 16,00,26,00,3A,A6,90,5F,3A,A7
2770 DATA 90,6F,2D,1D,CD,1D,BC,ED,4B,A4
2780 DATA 90,E5,16,0D,1E,05,0A,AE,77,03
2790 DATA 23,1D,20,FB,E1,CD,26,BC,E5,15
2800 DATA 20,EE,E1,C9,3A,8E,90,FE,00,C2
2810 DATA 4C,95,3E,01,32,8E,90,26,00,16
2820 DATA 00,3A,90,90,5F,3A,91,90,6F,2D
2830 DATA 1D,CD,1D,BC,CD,26,BC,CD,26,BC
2840 DATA CD,26,BC,01,17,A2,E5,16,07,1E
2850 DATA 05,0A,BE,20,11,23,03,1D,20,F7
2860 DATA E1,CD,26,BC,E5,15,20,ED,E1,C3
2870 DATA 4C,95,E1,3E,01,32,8F,90,C9,FE
2880 DATA 00,CA,29,92,47,3E,00,32,8E,90
2890 DATA 7B,C3,14,92,0E,03,06,FF,10,FE
2900 DATA 0D,20,F9,C3,AA,92,00,00,00,00
2910 DATA 00,00,00,00,00
2920
2930 CLS:RESTORE 2050:FOR i=&9200 TO &95
60:READ a$
2940 c=VAL("&" + a$):SOUND 129,c,5,5
2950 POKE i,c:NEXT i
2960 GOSUB 4010:GOTO 2980
2970 GOSUB 4220
2980 MODE 1:PEN 1:PAPER 0:LOCATE 6,6:PRI
NT"LEVEL?":CALL &BB03
2990 LOCATE 4,8:PRINT"0....BLUTIGER LAIE
"
3000 LOCATE 4,10:PRINT"1....ANFAENGER"
3010 LOCATE 4,12:PRINT"2....EXPERTE"
3020 LOCATE 4,14:PRINT"3....PROFI"
3030 LOCATE 4,16:PRINT"4....WAHNSINNIGER
"
```



```

3040 LOCATE 6,20:INPUT"EINGABE";lev
3050 IF lev>4 OR lev<0 THEN PRINT CHR$(7)
:GOTO 3040
3060 IF lev=0 THEN 11=4:12=255
3070 IF lev=1 THEN 11=3:12=255
3080 IF lev=2 THEN 11=2:12=255
3090 IF lev=3 THEN 11=1:12=255
3100 IF lev=4 THEN 11=1:12=1
3110 sc=0:live=3
3120 BORDER 1
3130 MODE 0:SPEED INK 6,6
3140 LOCATE 1,1:PEN 5:PRINT STRING$(20,C
HR$(143)):LOCATE 1,21:PRINT STRING$(20,C
HR$(143))
3150 FOR i=2 TO 15:LOCATE 1,i:PEN 5:PRIN
T CHR$(143);:PEN 6:PRINT CHR$(143)
3160 LOCATE 19,i:PRINT CHR$(143);:PEN 5:
PRINT CHR$(143):NEXT i
3170 LOCATE 3,2:PEN 6:PRINT STRING$(16,C
HR$(143))
3180 LOCATE 2,2:PEN 5:PAPER 6:PRINT CHR$(
212):LOCATE 19,2:PRINT CHR$(213)
3190 LOCATE 3,3:PEN 6:PAPER 0:PRINT CHR$(
212):LOCATE 18,3:PRINT CHR$(213)
3200 FOR i=0 TO 5:PEN 5:LOCATE 1,15+i:PR
INT STRING$(i,CHR$(143));CHR$(215)
3210 LOCATE 20-i,15+i:PRINT CHR$(214);ST

```

```

&BA
3420 POKE &90A4,z:POKE &90A5,y:POKE &90A
0,z:POKE &90A1,y
3430 POKE &909E,0:POKE &909F,0:POKE &909
4,0
3440 POKE &90BF,0
3450 y=INT (RND*50)+130:POKE &909B,y:POK
E &90A7,y
3460 IF x=1 THEN GOTO 3490
3470 y=50+INT (RND*40):POKE &9097,y:POKE
&90A6,y
3480 GOTO 3500
3490 y=110-INT (RND*40):POKE &9097,y:POKE
&90A6,y
3500 CALL &94C6
3510 CALL &9200
3520 IF PEEK(&90BF)=1 THEN 3540
3530 GOTO 3770
3540 'Explosion
3550 ENV 7,=11,1700:ENT -7,5,5,1,5,-5,1
3560 xk=4*PEEK(&9090):yk=2*PEEK(&9091)-2
0
3570 PRINT CHR$(23);CHR$(1);:MOVE 64,192
:DRAW xk,yk,1:MOVE 576,192:DRAW xk,yk,1
3580 SOUND 1,20,30,15,7,7:WHILE SQ(1)>12
7:WEND
3590 MOVE 64,192:DRAW xk,yk,1:MOVE 576,1

```

### *Nicht aufgeben! – Sie haben schon die Hälfte des Eintippens hinter sich!*

```

RING$(i,CHR$(143)):NEXT i
3220 PEN 6:1=1:FOR i=15 TO 20:PAPER 5:LO
CATE 1,i:PRINT CHR$(213);CHR$(143);:PAPE
R 0:PRINT CHR$(215)
3230 LOCATE 19-1,i:PRINT CHR$(214);:PAPE
R 5:PRINT CHR$(143);CHR$(212):1=1+1:NEXT
i
3240 PEN 6:PAPER 5:LOCATE 6,20:PRINT CHR
$(213);STRING$(8,CHR$(143));CHR$(212)
3250 LOCATE 1,15:PAPER 0:PEN 5:PRINT CHR
$(143)
3260 LOCATE 20,15:PRINT CHR$(143)
3270 PEN 8:LOCATE 3,14:PRINT STRING$(16,
CHR$(143))
3280 1=14:x=3:PEN 7:PAPER 6:FOR i=15 TO
19:LOCATE x,i:PRINT CHR$(213);STRING$(1,
CHR$(143));CHR$(212)
3290 1=1-2:x=x+1:IF i=16 THEN PEN 8
3300 NEXT i:PAPER 0
3310 PEN 9:FOR i=22 TO 25:LOCATE 1,i:PRI
NT STRING$(20,CHR$(143));:NEXT i:LOCATE
1,24
3320 PEN 1:PAPER 0
3330 PEN 3:LOCATE 1,25:PRINT STRING$(20,
CHR$(143));
3340 MOVE 1,16:DRAW 640,16,1
3350 PEN 2:PAPER 9:LOCATE 2,23:PRINT"SCD
RE":LOCATE 15,23:PRINT"LIVES"
3360 LOCATE 2,24:PRINT sc:LOCATE 16,24:P
RINT live
3370
3380 POKE &9090,80:POKE &9091,150
3390 CALL &927B:POKE &954D,11:POKE &954F
,12
3400 x=INT(RND*2):POKE &9095,x:POKE &909
C,0:POKE &909D,0
3410 IF x=0 THEN y=&A0:z=0 ELSE y=&A2:z=

```

```

92:DRAW xk,yk,1:PRINT CHR$(23);CHR$(0);
3600 ENV 6,=11,6000:ENT -6,1,1,4,1,-1,4,
1,1,1
3610 SOUND 129,700,100,15,6,6,30
3620 IF PEEK(&9094)=1 THEN 3740
3630 POKE &90A6,PEEK(&9097):POKE &90A7,P
EEK(&909B)
3640 POKE &90A4,PEEK(&90A0):POKE &90A5,P
EEK(&90A1)
3650 CALL &94C6
3660 FOR I=1 TO 30:POKE &90A4,&92:POKE &
90A5,&A4:CALL &94C6
3670 CALL &BD19
3680 POKE &90A4,&D3:POKE &90A5,&A4:CALL
&94C6
3690 CALL &BD19
3700 NEXT i
3710 IF PEEK(&9094)=1 THEN sc=sc+5 ELSE
sc=sc+100-(10*PEEK(&909C))
3720 PEN 2:PAPER 9:LOCATE 2,24:PRINT sc
3730 GOTO 3400
3740 POKE &90A6,PEEK(&90A8):POKE &90A7,P
EEK(&90A9)
3750 POKE &90A4,&49:POKE &90A5,&A2:CALL
&94C6
3760 GOTO 3660
3770 ENV 6,=11,7000:ENT -6,1,6,6,1,-6,1,
1,1,2,1,-1,2
3780 SOUND 129,200,120,15,6,6,15
3790 FOR i=1 TO 120:INK 5,INT(RND*20)+1:
INK 6,INT(RND*20)+1
3800 NEXT i
3810 INK 5,6:INK 6,4
3820 live=live-1:PEN 2:PAPER 9:LOCATE 16
,24:PRINT live
3830 IF live=0 THEN GOTO 3900
3840 POKE &90A6,PEEK(&90A8):POKE &90A7,P

```





```

EEK(&90A9)
3850 POKE &90A4,&49:POKE &90A5,&A2:CALL
&94C6
3860 PEN 15:PAPER 1:LOCATE 4,1:PRINT"SHI
P DESTROYED"
3870 ENV 6,=8,19:SOUND 129,300,220,15,6
3880 FOR i=1 TO 8:FOR l=0 TO 7:INK 15,1:
FOR k=1 TO 50:NEXT k,1,i
3890 LOCATE 4,1:PRINT"          ":GO
TO 3400
3900 PEN 15:PAPER 0:LOCATE 6,10:PRINT"GA
ME OVER"
3910 ENV 6,=8,19:ENT -6,1,3,2,1,-3,2,2,2
,2,2,-2,2
3920 SOUND 129,300,220,15,6,6
3930 FOR i=1 TO 8:FOR l=0 TO 7:INK 15,1:
FOR k=1 TO 50:NEXT k,1,i
3940 PEN 1:PAPER 0:MODE 1:IF sc>hsc THEN
GOTO 3990
3950 LOCATE 15,12:PRINT"HIGH SCORE":PEN
2:LOCATE 18,13:PRINT hsc
3960 LOCATE 20-LEN(nam$)/2,15:PEN 3:PRIN
T nam$
3970 LOCATE 15,20:PEN 1:PRINT"LAST SCORE
":PEN 2:LOCATE 18,21:PRINT sc
3980 GOTO 2970
3990 CALL &BB03:hsc=sc:LOCATE 12,10:PRIN
T"DEIN NAME BITTE":LOCATE 10,12:INPUT na
m$
4000 CLS:GOTO 3950
4010 SYMBOL 200,0,126,126,90,90,24,24,24
4020 SYMBOL 201,24,24,24,24,24,60,60,0
4030 SYMBOL 202,0,252,252,102,102,102,10
2,124
4040 SYMBOL 203,124,108,108,102,102,226,
226,0
4050 SYMBOL 204,0,126,126,24,24,24,24,24
4060 SYMBOL 205,24,24,24,24,24,126,126,0
4070 SYMBOL 206,0,56,56,108,108,198,198,
198
4080 SYMBOL 207,198,198,198,108,108,56,5
6,0
4090 SYMBOL 208,0,198,198,230,230,246,24
6,222
4100 SYMBOL 209,222,206,206,198,198,198,
198,0
4110 MODE 0:PEN 1:BORDER 0:PAPER 0:INK 0
,0
4120 LOCATE 5,6
4130 PRINT CHR$(200);" ";CHR$(202);" ";C
HR$(204);" ";CHR$(200);" ";CHR$(206);" "
;CHR$(208)
4140 LOCATE 5,7
4150 PRINT CHR$(201);" ";CHR$(203);" ";C
HR$(205);" ";CHR$(201);" ";CHR$(207);" "
;CHR$(209)
4160 LOCATE 6,9:PEN 2:PRINT"(c) 1986"
4170 LOCATE 2,11:PEN 1:PRINT"OTTOFRIED S
CHMIDT"
4180 PEN 3:LOCATE 5,15:PRINT"PRESS FIRE
"
4190 PEN 12:LOCATE 1,4:PRINT CHR$(150);S
TRING$(18,CHR$(154));CHR$(156)
4200 LOCATE 1,17:PRINT CHR$(147);STRING$
(18,CHR$(154));CHR$(153)
4210 FOR i=5 TO 16:LOCATE 1,i:PRINT CHR$
(149):LOCATE 20,i:PRINT CHR$(149):NEXT i
4220 PEN 1

```

```

4230 ENV 1,=11,7000:ENV 2,1,15,1,15,-1,5
:ENT -1,3,0.75,6,3,-0.75,6:ENT -2,2,2,4,
2,-2,4,1,1,7,1,-1,7
4240 b=4:x=478:RESTORE 4310
4250 READ a:IF a=0 THEN RESTORE 4320:REA
D a:IF b=4 THEN b=2 ELSE b=4
4260 IF a<>1 THEN SOUND 130,a*2,90,7,1,1
:SOUND 132,a*b,160,7,1,2
4270 SOUND 1,x,24,10,2,2:IF x=478 THEN x
=379 ELSE x=478
4280 IF JOY(0) AND 16 THEN RETURN
4290 WHILE SQ(1)>127:WEND
4300 GOTO 4250
4310 DATA 1,1,1,1,119,1,1,80,1,1,1,1,1,
1,119,1,1,80,1,1,1,1,1,1
4320 DATA 119,89,80,71,80,1,95,1
4330 DATA 1,119,89,80,71,80,1
4340 DATA 1,1,119,89,80,71,80,1,95,1
4350 DATA 1,95,89,95,119,119,1
4360 DATA 1,1,60,63,71,80,63,1,80,71,1,8
9,80
4370 DATA 1,60,63,71,80,63,1,80,71,1,89,
80
4380 DATA 1,95,89,95,119,119
4390 DATA 1,1,1,1,1,1,1,1,0

```

## TRITON-Checksummen

ZEILENR.:	SUMMEN				
1-	2: 24952	1040-	1050: 11103	2140-	2150: 51578
3-	4: 26032	1060-	1070: 21555	2160-	2170: 52376
5-	6: 16520	1080-	1090: 21667	2180-	2190: 53464
7-	10: 45528	1100-	1110: 21796	2200-	2210: 52906
20-	30: 41448	1120-	1130: 21762	2220-	2230: 51419
40-	50: 37332	1140-	1150: 21755	2240-	2250: 52869
60-	70: 14554	1160-	1170: 21551	2260-	2270: 52046
80-	90: 21447	1180-	1190: 11037	2280-	2290: 51312
100-	110: 21421	1200-	1210: 21469	2300-	2310: 51567
120-	130: 21304	1220-	1230: 21469	2320-	2330: 53021
140-	150: 21304	1240-	1250: 21975	2340-	2350: 51408
160-	170: 21304	1260-	1270: 21579	2360-	2370: 51547
180-	190: 21304	1280-	1290: 21469	2380-	2390: 51300
200-	210: 11037	1300-	1310: 21381	2400-	2410: 51387
220-	230: 21525	1320-	1330: 11037	2420-	2430: 52050
240-	250: 21733	1340-	1350: 21304	2440-	2450: 52389
260-	270: 21304	1360-	1370: 21460	2460-	2470: 51307
280-	290: 21304	1380-	1390: 21616	2480-	2490: 51795
300-	310: 21304	1400-	1410: 21629	2500-	2510: 51540
320-	330: 21304	1420-	1430: 21629	2520-	2530: 51385
340-	350: 11037	1440-	1450: 21382	2540-	2550: 51455
360-	370: 21499	1460-	1470: 770	2560-	2570: 52323
380-	390: 21712	1480-	1490: 22351	2580-	2590: 53333
400-	410: 21397	1500-	1510: 24552	2600-	2610: 51847
420-	430: 21304	1520-	1530: 25660	2620-	2630: 50962
440-	450: 21304	1540-	1550: 26869	2640-	2650: 51684
460-	470: 21304	1560-	1570: 24587	2660-	2670: 51701
480-	490: 11037	1580-	1590: 23404	2680-	2690: 51304
500-	510: 21381	1600-	1610: 11037	2700-	2710: 54397
520-	530: 21508	1620-	1630: 25678	2720-	2730: 54096
540-	550: 21634	1640-	1650: 24655	2740-	2750: 52906
560-	570: 21515	1660-	1670: 25765	2760-	2770: 54125
580-	590: 21304	1680-	1690: 25630	2780-	2790: 52708
600-	610: 21304	1700-	1710: 24539	2800-	2810: 51591
620-	630: 11037	1720-	1730: 24574	2820-	2830: 54266
640-	650: 21381	1740-	1750: 23079	2840-	2850: 51193
660-	670: 21461	1760-	1770: 50128	2860-	2870: 53795
680-	690: 21513	1780-	1790: 39060	2880-	2890: 51992
700-	710: 21535	1800-	1810: 1144	2900-	2910: 30938
720-	730: 21384	1820-	1830: 41899	2920-	2930: 36088
740-	750: 21304	1840-	1850: 31782	2940-	2950: 42018
760-	770: 11037	1860-	1870: 71163	2960-	2970: 4986
780-	790: 21413	1880-	1890: 31782	2980-	2990: 56278
800-	810: 21535	1900-	1910: 27487	3000-	3010: 34097
820-	830: 21627	1920-	1930: 1872	3020-	3030: 37244
840-	850: 21591	1940-	1950: 27904	3040-	3050: 78577
860-	870: 21456	1960-	1970: 52747	3060-	3070: 81678
880-	890: 21304	1980-	1990: 24734	3080-	3090: 81638
900-	910: 11037	2000-	2010: 37415	3100-	3110: 48257
920-	930: 21402	2020-	2030: 1189	3120-	3130: 3393
940-	950: 21565	2040-	2050: 25629	3140-	3150: 32171
960-	970: 21690	2060-	2070: 54736	3160-	3170: 75544
980-	990: 21790	2080-	2090: 53978	3180-	3190: 102545
1000-	1010: 21698	2100-	2110: 52002	3200-	3210: 73660
1020-	1030: 21444	2120-	2130: 52106		

Checksummen	Checksummen	Checksummen	Checksummen	Checksummen
3220- 3230: 79174 3240- 3250: 28089 3260- 3270: 33024 3280- 3290: 120358 3300- 3310: 41671 3320- 3330: 26996 3340- 3350: 53111 3360- 3370: 32959 3380- 3390: 42145 3400- 3410: 85684 3420- 3430: 26748	3440- 3450: 4645 3460- 3470: 75224 3480- 3490: 65852 3500- 3510: 2692 3520- 3530: 14625 3540- 3550: 32039 3560- 3570: 124230 3580- 3590: 33979 3600- 3610: 39317 3620- 3630: 47086 3640- 3650: 36396	3660- 3670: 61921 3680- 3690: 27991 3700- 3710: 29913 3720- 3730: 15186 3740- 3750: 60572 3760- 3770: 47048 3780- 3790: 38323 3800- 3810: 4060 3820- 3830: 79184 3840- 3850: 60572 3860- 3870: 51765 3880- 3890: 19229 3900- 3910: 65507	3920- 3930: 16835 3940- 3950: 99632 3960- 3970: 98125 3980- 3990: 50794 4000- 4010: 18168 4020- 4030: 39818 4040- 4050: 41568 4060- 4070: 45352 4080- 4090: 60054 4100- 4110: 41546 4120- 4130: 11455 4140- 4150: 11667 4160- 4170: 42747	4180- 4190: 80872 4200- 4210: 93927 4220- 4230: 8529 4240- 4250: 81396 4260- 4270: 68623 4280- 4290: 18386 4300- 4310: 8239 4320- 4330: 26228 4340- 4350: 32681 4360- 4370: 62317 4380- 4390: 17767 GESAMTCHECKSUMME: 33508
Checksummen	Checksummen	Checksummen	Checksummen	Checksummen

## Neu: Sonderzeichen werden durch Klartext ersetzt!

### Eingabe-Hinweise für Commodore C-16/116-Benutzer!

Um Ihnen die Eingabe unserer Listings zu erleichtern, werden zukünftig all C-16/116-Listings ohne Steuerzeichen abgedruckt. Alle Steuerzeichen werden in unseren LISTINGS durch Klartext, welcher die Taste kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin.

#### Tastenbezeichnungen für Steuercodes: C-16/116 u. Plus 4 (TRONIC-NORM)

SPACE	LEERZEICHEN	LEERTASTE (GROSSESTE TASTE)	ORNG	ORANGE	Commodore-Taste + 1
F1	FUNKTIONSTASTE	FUNKTIONSTASTE F1	BRN	BRAUN	Commodore-Taste + 2
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F3	YL.GRN	GELB-GRÜN	Commodore-Taste - 3
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F5	DINK	ROSA	Commodore-Taste + 4
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F7	BL.GRN	Blau-Grün	Commodore-Taste + 5
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F2	L.BLU	Hellblau	Commodore-Taste + 6
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F4	D.BLU	Dunkelblau	Commodore-Taste + 7
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F6	L.GRN	Hellgrün	Commodore-Taste + 8
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8			
BLACK	SCHWARZ	CONTROL-TASTE & 1	DOWN	Cursor unten	Taste neben INST DEL
WHITE	WEISS	CONTROL-TASTE & 2	UP	Cursor hoch	3. Taste neben DOWN von rechts oben
RED	ROT	CONTROL-TASTE & 3	RIGHT	Cursor rechts	4. Taste von rechts oben
CYAN	TUERKIS	CONTROL-TASTE & 4	LEFT	Cursor links	5. Taste von rechts oben
PURPLE	PURPUR	CONTROL-TASTE & 5	CLR	Schirm löschen	Taste über Return zusammen mit Shift
GREEN	GRÜN	CONTROL-TASTE & 6	HOME	Cursor in Ecke	Taste über Return
BLUE	BLAU	CONTROL-TASTE & 7	DEL	Delete	Taste ganz rechts oben
YELLOW	GELB	CONTROL-TASTE & 8	Flash on	Blinken ein	Controll-Taste + ,
RVSON	INVERSE EIN	CONTROL-TASTE & 9	Flash off	Blinken aus	Controll-Taste + .
RVSOFF	INVERSE AUS	CONTROL-TASTE & 0			
CTRL	CONTROL	CONTROL-TASTE ZUSAMMEN			
		MIT NACHFOLGENDEN ZEICHEN. Z.B. CTRLA ENTSPRICHT CTRL & A)			

#### Tastenbezeichnungen für Steuercodes

#### VC-20 und Commodore 64

(TRONIC-NORM)

DOWN	CURSOR UNTEN	TASTE NEBEN RECHTEM SHIFT	CYAN	TUERKIS	CONTROL-TASTE & 4
UP	CURSOR HOCH	SHIFT-TASTE & TASTE NEBEN RECHTEM SHIFT	PURPLE	PURPUR	CONTROL-TASTE & 5
CLR	CLEAR SCHIRM	SHIFT-TASTE & 2. TASTE GANZ RECHTS OBEN	GREEN	GRUEN	CONTROL-TASTE & 6
INST	EINFUEGEN	SHIFT-TASTE & TASTE GANZ RECHTS OBEN	BLUE	BLAU	CONTROL-TASTE & 7
HOME	CURSOR IN ECKE	2. TASTE VON GANZ RECHTS OBEN	YELLOW	GELB	CONTROL-TASTE & 8
DEL	DELETE	TASTE GANZ RECHTS OBEN	RVSON	INVERSE EIN	CONTROL-TASTE & 9
RIGHT	CURSOR RECHTS	TASTE GANZ RECHTS UNTEN	RVSOFF	INVERSE AUS	CONTROL-TASTE & 0
LEFT	CURSOR LINKS	SHIFT-TASTE & TASTE UNTEN RECHTS	ORANGE	ORANGE	COMMODORE-TASTE & 1
SPACE	LEERZEICHEN	LEERTASTE (GROESSTE TASTE)	BROWN	BRAUN	COMMODORE-TASTE & 2
F1	FUNKTIONSTASTE	FUNKTIONSTASTE F1	LIG.RED	HELLROT	COMMODORE-TASTE & 3
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F3	DGREY	DUNKELGRAU	COMMODORE-TASTE & 4
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F5	MGREY	MITTELGRAU	COMMODORE-TASTE & 5
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F7	LIG.GREEN	HELLGRUEN	COMMODORE-TASTE & 6
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F2	LIG.BLUE	HELLBLAU	COMMODORE-TASTE & 7
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F4	HGREY	HELLGRAU	COMMODORE-TASTE & 8
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F6	CTRL	CONTROL	CONTROL-TASTE ZUSAMMEN MIT DEM NACHFOLGENDEN ZEICHEN. (Z.B. CTRLA ENTSPRICHT CTRL & A)
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8			
BLACK	SCHWARZ	CONTROL-TASTE & 1	FLASH ON	BLINKEN EIN	CONTROL-TASTE & ,
WHITE	WEISS	CONTROL-TASTE & 2	FLASH OFF	BLINKEN AUS	CONTROL-TASTE & .
RED	ROT	CONTROL-TASTE & 3			

Achtung: Einige Listings enthalten das Zeichen "-", dieses Zeichen entspricht dem Pfeil nach links (Taste ganz links oben)  
das Zeichen "^" steht für den Pfeil nach oben,  
das Zeichen "\\_" steht für das (engl.) Pfund-Symbol (Taste neben Restore)

# Topprogramm Boulder Dash Atari 800XL/130XE

**Jetzt können Sie über diese bekannte  
C-64-Version auch auf Ihrem Atari  
Computer verfügen!**

In diesem Spiel übernehmen Sie die Rolle eines Steinbeißers, der die schwierige Aufgabe hat, so viele Diamanten zu sammeln, wie in der linken oberen Ecke angezeigt werden. In der rechten oberen Ecke befindet sich die Punkteanzeige, in der Mitte eine Uhr und der Zähler für die bereits eingesammelten Diamanten. Während Ihrer Aktion müssen Sie sich vor folgenden Dingen schützen:

## Steine

Steine können Sie nach links oder rechts verschieben. Gräbt man unter ihnen die Erde weg, so fallen diese solange bis sie wieder auf einen festen Gegenstand treffen. Achten Sie immer darauf, daß sie nicht aus Unachtsamkeit von einem Stein erschlagen werden.

## Quadratisches Monster

Berührt man eines von ihnen, hat dies den Verlust eines Lebens zur Folge.

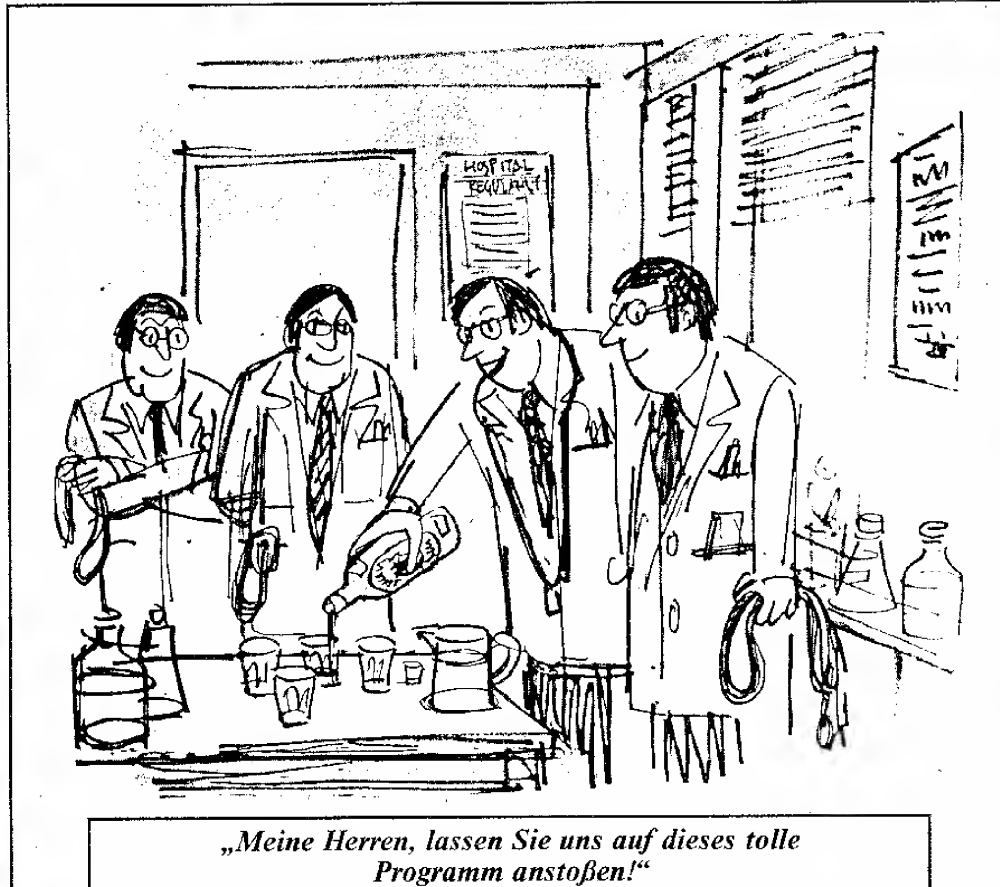
## Schmetterlinge

Die Schmetterlinge haben die Eigenschaft, daß sie sich in Diamanten verwandeln, wenn sie zerstört werden. Eine Berührung mit ihnen ist ebenfalls ungesund.

## Schwabbelnde Masse

Die Masse wächst über leere Felder oder Erde. Hat sie keine Möglichkeit mehr zu wachsen, dann verwandelt sie sich in Diamanten. Wird sie zu groß, verwandelt sie sich zu Stein. Berührt man sie mit der eigenen Figur, hat es keine Folgen. Für die Monster allerdings ist eine Berührung der Masse tödlich (muß in manchen Situationen angewandt werden, um Monster zu zerstören). Sie können sich der Monster ebenfalls erwehren, indem Sie sie unter einem Stein zerdrücken.

Haben Sie Probleme bei der Lösung eines Levels, dann denken Sie immer daran, daß es möglich ist, aus Schmetterlingen Diamanten zu gewinnen. Ebenfalls ist es manchmal nötig, einige Mauern zu sprengen, indem man Monster, die sich in unmittelbarer Nähe befinden mit einem Stein erschlägt. Da die Monster dabei ihre Umgebung freisprengen, ist es somit möglich Ziegelmauern zu durchbrechen. Weil



*„Meine Herren, lassen Sie uns auf dieses tolle  
Programm anstoßen!“*

man immer nur einen Teil des Spielfeldes auf einmal sehen kann, lohnt es sich oft, wenn man erst einmal ein wenig herumläuft um sich einen Gesamtüberblick über das Spielfeld zu verschaffen.

Ist man in eine Lage geraten aus der es keinen Ausweg mehr gibt, so kann man diesen Level durch Drücken der „ESC“ Taste noch einmal von vorn beginnen.

Durch Drücken von „start“ kommt man wieder zum Titelbild.

Im Titelbild lassen sich auch die verschiedenen Level anwählen, indem man den Joystick nach links oder rechts drückt. Durch Drücken des Feuerknopfes beginnt das Spiel.

## Kurzbeschreibung der Programme

### Teil 1 und 2

Diese beiden Programme erzeugen jeweils ein Datenfile auf Kassette, aus denen später das Hauptprogramm zusammengesetzt wird. Die Werte in den Datas sind iden-

tisch mit denen der Diskettenversion von Boulder Dash, die Ihnen bereits vorliegt.

### Teil 3 - CONNECT.BAS

Mit diesem Programm werden die beiden Datenfiles 1 und 2 zu einem ladbaren File verbunden. Zuerst werden die Inhalte der beiden Files in den Speicher eingelesen und dann nach der Meldung „Programm speichern“ wieder auf Kassette in einem File abgelegt. Vor dem Hauptprogramm sollte man etwas Platz auf der Kassette lassen und das Programm DASHLOAD.BAS dort abspeichern, falls man das nicht schon getan hat.

### Teil 4 - DASHLOAD.BAS

Dieses Programm ist nötig, um das aus den Datenfiles 1 und 2 zusammengesetzte Hauptprogramm einzuladen. Man lädt also zuerst dieses Programm und startet es. Dann wird man aufgefordert zum Hauptprogramm vorzuspulen und die Play-Taste am Recorder zu drücken. Drückt man nur die Return-Taste, so wird der Ladevorgang



begonnen. Nach dem etwa 12minütigen Ladevorgang startet sich das Programm dann von selbst.

## Hinweise zur Erzeugung des Programmes

Diskettenbenutzer geben zuerst Teil 1 ein und speichern diesen unter dem Namen DASHGEN1.BAS ab. Nun den zweiten Teil eingeben und unter DASHGEN2.BAS abspeichern.

Nun wird Teil 1 geladen und mit RUN gestartet. Es sollte dabei eine, bis auf Teil 1 und Teil 2, leere Diskette verwendet werden. Während des etwa 9minütigen Generierens wird Teil 2 nachgeladen und ausgeführt.

Ist dies planmäßig verlaufen, müßte sich

nun das File DASH.COM auf der Diskette befinden.

Tritt ein Fehler auf, so wird die Zeile in der der Fehler erkannt wurde, angezeigt. Nun müßte der Fehler im näheren Umfeld dieser Zeile zu finden sein. Ist dieser behoben so wird das Programm erneut mit RUN gestartet. Das Spiel kann nun im DOS über die L-Funktion geladen und gestartet werden.

Kassettenbenutzer verwenden am besten drei Kassetten.

Auf der Kassette, auf der sich das Spiel später befinden soll wird zuerst Teil 4 abgespeichert.

Nun die 1. Hilfskassette einlegen und Teil 3 abspeichern.

Um nun die beiden erforderlichen Datenfiles zu erzeugen ist es notwendig die geänderten Einleseschleifen zu verwenden (Teil 1b und Teil 2b), die Datenzeilen werden nicht verändert, sie sind bei Kassette

und Diskette identisch.

Diese beiden Programme werden nun nacheinander auf der 2. Hilfskassette abgespeichert. Nun die 2. Hilfskassette zurückspulen, Teil 1 laden und mit RUN starten. Das Datenfile wird jetzt auf die 1. Hilfskassette geschrieben. Nach Beendigung des Schreibvorganges wird Teil 2 von der 2. Hilfskassette nachgeladen und gestartet. Anschließend wird das zweite Datenfile auf der 1. Hilfskassette abgespeichert. Nun die 1. Hilfskassette zurückspulen, Teil 3 laden und mit RUN starten. Anschließend werden die beiden Datenfiles nachgeladen und zusammen auf der Spielkassette abgespeichert (einfach die Anweisungen des Programms befolgen).

Wollen Sie jetzt spielen, so müssen Sie nur die Spielkassette einlegen, das erste Programm laden (mit CLOAD) und mit RUN starten, nach einer 12minütigen Ladezeit startet das Programm automatisch.

**Achtung Kassettenbenutzer! Kassettenbenutzer müssen in Teil 1 und in Teil 2 die ursprünglichen Zeilen von 1 bis einschließlich 140 löschen und durch die Teile 1b und 2b ersetzen.**

## Listing: BOULDER DASH

```

1 REM *****
2 REM *
3 REM *      BOULDER DASH      *
4 REM *      TEIL 1          *
5 REM *      ERZEUGT 'D:DASH.ZPS' *
6 REM *      (C) 1986        *
7 REM *      VON OLIVER CYRANKA *
8 REM *
9 REM *****
10 OPEN #5,B,0,"D:DASH.ZPS":TRAP 60:POKE
    559,0
20 FOR I=300 TO 688:S=0:FOR J=1 TO 25:RE
AD A:PUT #5,A:S=S+A:NEXT J:READ A:IF A=S
    THEN NEXT I:GOTO 50
30 ? "DATENFEHLER IN ZEILE ";I
40 POKE 559,34:CLOSE #5:END
50 CLOSE #5:RUN "D:DASHGEN2.BAS"
60 ER=PEEK(195)
70 IF ER=8 THEN ? "FEHLER IN ZEILE ";I:G
    OTO 40
80 ? "FEHLER ";ER;" IN ZEILE ";PEEK(186)
+PEEK(187)*256:GOTO 40
120 REM
130 REM **** DATEN FUER DAS FILE ****
140 REM
300 DATA 255,255,0,91,31,156,100,2,0,0,2
0,2,26,1,28,3,88,1,90,3,98,1,100,2,0,135
3
301 DATA 0,20,2,44,1,46,3,72,1,74,3,98,1
,100,2,0,0,20,2,34,1,36,3,56,1,620
302 DATA 58,7,60,1,62,7,78,1,80,3,98,1,1
00,2,0,0,20,2,46,1,48,3,54,1,56,789
303 DATA 7,58,8,60,5,62,8,64,7,90,1,92,3
,98,1,100,2,0,0,20,2,28,1,30,3,750
304 DATA 52,1,54,7,56,8,58,5,60,1,62,5,6
4,8,64,7,76,1,78,3,98,1,100,2,0,873
305 DATA 0,20,2,32,1,34,3,50,1,52,7,54,8
,56,5,58,1,60,4,62,1,64,5,66,8,654

```

```

306 DATA 68,7,94,1,96,3,98,1,100,2,0,0,2
0,2,38,1,40,3,50,1,52,7,54,8,56,802
307 DATA 5,58,1,60,4,62,1,64,5,66,8,68,7
,76,1,78,3,98,1,100,2,0,0,20,2,790
308 DATA 44,1,48,3,52,1,54,7,56,8,58,5,6
0,1,62,5,64,8,66,7,70,1,72,3,98,854
309 DATA 1,100,2,0,0,20,2,48,1,50,3,54,1
,56,7,58,8,60,5,62,8,64,7,90,1,708
310 DATA 92,3,98,1,100,2,0,0,20,2,24,1,2
6,3,56,1,58,7,60,1,62,7,66,1,68,759
311 DATA 3,98,1,100,2,0,0,20,2,26,1,28,3
,32,1,36,7,56,1,58,3,74,1,76,3,632
312 DATA 82,1,86,7,98,1,100,2,0,0,20,2,3
0,1,32,7,34,8,36,5,38,7,46,1,50,694
313 DATA 3,66,1,68,3,80,1,82,7,84,8,86,5
,88,7,98,1,100,2,0,0,20,2,28,1,841
314 DATA 30,7,32,8,36,5,38,8,40,7,70,1,7
2,3,78,1,80,7,82,8,86,5,88,8,90,890
315 DATA 7,96,1,98,3,100,2,0,0,20,2,26,1
,28,7,30,8,32,5,36,1,38,5,40,8,594
316 DATA 42,7,50,1,52,3,76,1,78,7,80,8,8
2,5,86,1,88,5,90,8,92,7,98,1,100,1068
317 DATA 2,0,0,20,2,28,1,32,5,36,4,40,5,
64,1,66,3,78,1,82,5,86,4,90,5,660
318 DATA 98,1,100,2,0,0,20,2,22,3,26,1,2
8,7,30,8,32,5,36,1,38,5,40,8,42,555
319 DATA 7,76,1,78,7,80,8,82,5,86,1,88,5
,90,8,92,7,98,1,100,2,0,0,20,2,944
320 DATA 28,1,30,7,32,8,36,5,38,8,40,7,4
8,1,50,3,62,1,64,3,78,1,80,7,82,720
321 DATA 8,86,5,88,8,90,7,98,1,100,2,0,0
,20,2,30,1,32,7,34,8,36,5,38,7,713
322 DATA 42,1,44,3,60,1,62,3,80,1,82,7,8
4,8,86,5,88,7,98,1,100,2,0,0,20,885
323 DATA 2,24,1,26,3,32,1,36,7,40,1,42,3
,50,1,52,3,60,1,62,3,76,1,78,3,608
324 DATA 82,1,86,7,98,1,100,2,0,0,20,2,3
0,1,32,3,70,1,72,3,98,1,100,2,0,812

```



325 DATA 0,20,2,88,1,92,3,98,1,100,2,0,0,  
100,2,0,0,0,0,0,0,0,0,0,0,509  
326 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,25  
327 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,25  
328 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,25  
329 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
1,1,1,1,1,1,1,1,1,1,25  
330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
0,0,0,0,0,0,0,0,100  
331 DATA 2,0,0,20,2,24,1,26,8,30,1,32,3,  
44,1,46,3,58,1,60,2,70,1,72,3,510  
332 DATA 88,1,90,3,98,1,100,2,0,0,20,2,2,  
4,1,26,8,56,1,58,3,90,1,92,3,98,866  
333 DATA 1,100,2,0,0,20,2,24,1,26,8,34,1,  
36,3,48,1,50,3,70,1,72,3,94,1,601  
334 DATA 96,3,98,1,100,2,0,0,20,2,24,1,2,  
6,8,38,1,40,3,76,1,78,3,98,1,100,820  
335 DATA 2,0,0,20,2,24,1,26,3,50,1,52,3,  
80,1,82,3,98,1,100,2,0,0,20,2,573  
336 DATA 28,1,30,3,92,1,94,3,98,1,100,2,  
0,0,20,2,24,1,26,3,34,1,84,7,96,751  
337 DATA 1,98,3,100,2,0,0,20,2,28,1,30,3,  
34,1,36,7,82,6,84,7,98,1,100,2,746  
338 DATA 0,0,20,2,34,1,36,7,40,6,48,7,50,  
6,52,7,56,6,58,7,60,6,66,7,70,652  
339 DATA 6,78,7,82,6,84,7,98,1,100,2,0,0,  
20,2,34,1,36,7,40,6,42,7,50,6,722  
340 DATA 52,7,54,6,58,7,60,6,62,7,64,6,6,  
8,7,70,6,72,7,82,6,84,7,98,1,100,997  
341 DATA 2,0,0,20,2,34,1,36,7,40,6,46,7,  
50,6,58,7,60,6,62,7,66,6,68,7,604  
342 DATA 70,6,76,7,82,6,84,7,98,1,100,2,  
0,0,20,2,34,1,36,7,40,6,42,7,50,784  
343 DATA 6,54,7,56,6,58,7,60,6,62,7,64,6,  
68,7,70,6,72,7,82,6,84,7,98,1,907  
344 DATA 100,2,0,0,20,2,34,1,36,7,40,6,4,  
8,7,50,6,52,7,56,6,58,7,60,6,66,677  
345 DATA 7,70,6,78,7,82,6,84,7,98,1,100,  
2,0,0,20,2,28,1,30,3,34,1,36,7,710  
346 DATA 82,6,84,7,88,1,90,3,98,1,100,2,  
0,0,20,2,34,1,84,7,98,1,100,2,0,911  
347 DATA 0,20,2,22,1,24,3,84,1,86,3,98,1,  
100,2,0,0,20,2,28,1,30,3,46,1,578  
348 DATA 48,3,56,1,58,3,72,1,74,3,88,1,9,  
0,3,98,1,100,2,0,0,20,2,40,1,42,807  
349 DATA 3,94,1,96,3,98,1,100,2,0,0,20,2,  
58,1,60,3,98,1,100,2,0,0,20,2,765  
350 DATA 44,1,46,3,72,1,74,3,98,1,100,2,  
0,0,20,2,98,1,100,2,0,0,100,2,0,770  
351 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,  
20,2,40,9,42,7,90,1,92,7,412  
352 DATA 98,1,100,2,0,0,20,2,22,9,38,8,4,  
0,9,42,7,44,1,46,3,50,1,84,3,90,720  
353 DATA 1,92,7,98,1,100,2,0,0,20,2,22,9,  
24,8,36,1,38,8,40,9,42,7,90,1,658  
354 DATA 92,7,98,1,100,2,0,0,20,2,22,9,2,  
4,8,36,1,38,8,40,9,42,7,90,1,92,749  
355 DATA 7,98,1,100,2,0,0,20,2,22,9,24,8,  
36,1,38,8,40,9,42,7,90,1,92,7,664  
356 DATA 98,1,100,2,0,0,20,2,22,9,24,8,3,  
6,1,38,8,40,9,42,7,90,1,92,7,98,755

357 DATA 1,100,2,0,0,20,2,22,9,24,8,36,1,  
38,8,40,9,42,7,90,1,92,7,98,1,658  
358 DATA 100,2,0,0,20,2,22,9,24,8,36,1,3,  
8,8,40,9,42,7,90,1,92,7,98,1,100,757  
359 DATA 2,0,0,20,2,22,9,38,8,40,9,42,7,  
90,1,92,7,98,1,100,2,0,0,20,2,612  
360 DATA 40,9,42,7,90,1,92,7,98,1,100,2,  
0,0,20,2,42,7,90,1,92,7,98,1,100,949  
361 DATA 2,0,0,20,2,90,1,92,7,98,1,100,2,  
0,0,20,2,90,1,92,7,98,1,100,2,828  
362 DATA 0,0,20,2,90,1,92,7,98,1,100,2,0,  
0,20,2,90,1,92,7,98,1,100,2,0,826  
363 DATA 0,20,2,50,1,92,7,98,1,100,2,0,0,  
20,2,46,1,96,5,98,1,100,2,0,0,744  
364 DATA 20,2,46,1,52,5,54,2,58,5,60,2,6,  
4,5,66,2,70,5,72,2,76,5,78,2,82,836  
365 DATA 5,84,2,88,5,90,2,94,5,98,1,100,  
2,0,0,20,2,32,1,34,3,46,1,52,5,772  
366 DATA 54,2,58,5,60,2,64,5,66,2,68,8,7,  
0,5,72,2,74,8,76,5,78,2,80,8,82,956  
367 DATA 5,84,2,88,5,90,2,94,5,98,1,100,  
2,0,0,20,2,46,1,52,5,54,2,58,5,821  
368 DATA 60,2,64,5,66,2,70,5,72,2,76,5,7,  
8,2,82,5,84,2,88,5,90,2,94,5,98,1064  
369 DATA 1,100,2,0,0,20,2,46,1,94,5,98,1,  
100,2,0,0,100,2,0,0,100,2,0,0,676  
370 DATA 20,2,32,1,34,3,54,1,56,4,60,5,6,  
2,3,76,1,78,3,98,1,100,2,0,0,20,716  
371 DATA 2,24,1,26,3,28,1,30,4,34,3,42,1,  
44,3,56,1,58,3,74,1,76,3,98,1,617  
372 DATA 100,2,0,0,20,2,58,1,60,3,66,1,6,  
8,3,72,1,74,3,76,4,86,1,88,3,92,884  
373 DATA 1,94,4,98,1,100,2,0,0,20,2,38,1,  
40,3,48,1,50,4,82,1,84,3,96,1,774  
374 DATA 98,3,100,2,0,0,20,2,80,7,82,5,9,  
8,7,100,2,0,0,20,2,36,1,38,4,44,751  
375 DATA 1,46,3,48,1,50,7,60,1,62,3,74,1,  
76,7,86,1,88,7,94,1,96,3,98,1,915  
376 DATA 100,2,0,0,20,2,22,3,24,1,26,4,2,  
8,1,30,3,48,1,50,7,62,1,64,4,68,571  
377 DATA 1,70,3,74,1,76,7,78,3,82,1,84,3,  
86,1,88,7,98,1,100,2,0,0,20,2,888  
378 DATA 48,1,50,7,54,1,56,3,86,1,88,7,9,  
0,3,96,1,98,4,100,2,0,0,20,2,32,850  
379 DATA 1,34,3,64,1,66,3,74,1,76,7,82,1,  
84,3,86,1,88,7,98,1,100,2,0,0,883  
380 DATA 20,2,44,1,46,3,48,1,50,7,54,1,5,  
6,4,70,1,72,3,74,1,88,7,94,1,96,844  
381 DATA 3,98,1,100,2,0,0,20,2,24,1,26,3,  
40,1,42,4,48,1,50,7,60,1,62,3,599  
382 DATA 80,1,82,7,88,1,90,3,98,1,100,2,  
0,0,20,2,36,1,38,3,48,1,50,7,54,813  
383 DATA 1,56,3,76,1,78,4,80,1,82,7,84,1,  
86,4,98,1,100,2,0,0,20,2,26,7,820  
384 DATA 28,1,92,7,94,1,98,1,100,2,0,0,2,  
0,2,22,3,34,1,36,3,46,1,48,4,60,704  
385 DATA 1,62,7,76,1,78,3,98,1,100,2,0,0,  
20,2,22,3,24,4,40,1,42,3,54,1,645  
386 DATA 56,3,60,1,62,7,68,1,70,3,94,1,9,  
6,3,98,1,100,2,0,0,20,2,22,3,26,799  
387 DATA 1,28,3,44,1,48,3,60,1,62,7,64,3,  
74,1,76,4,84,1,86,3,98,1,100,2,855  
388 DATA 0,0,20,2,24,3,60,1,62,7,78,1,80,  
3,98,1,100,2,0,0,20,2,36,1,38,639



# ATARI



389 DATA 3,54,1,56,3,60,1,62,7,86,1,88,3,98,1,100,2,0,0,20,2,30,1,32,3,714  
390 DATA 42,1,44,4,68,1,70,3,84,1,86,4,98,1,100,2,0,0,20,2,30,1,34,3,76,775  
391 DATA 1,78,3,98,1,100,2,0,0,20,2,46,1,48,3,78,1,80,3,94,1,96,3,98,1,858  
392 DATA 100,2,0,0,100,2,0,0,100,2,0,0,2,0,2,24,1,28,3,70,1,72,3,98,1,100,729  
393 DATA 2,0,0,20,2,56,1,58,3,98,1,100,2,0,0,20,2,38,7,40,1,42,3,58,1,555  
394 DATA 62,3,80,1,98,7,100,2,0,0,20,2,2,2,4,24,1,36,5,38,7,48,1,50,3,52,666  
395 DATA 8,80,1,82,7,94,5,96,1,98,4,100,2,0,0,20,2,22,4,24,1,26,8,36,5,726  
396 DATA 38,7,72,1,76,3,80,1,82,7,84,8,94,5,96,1,98,4,100,2,0,0,20,2,28,909  
397 DATA 7,30,1,38,7,44,1,50,3,62,1,64,3,70,1,72,8,80,1,88,7,90,1,98,7,834  
398 DATA 100,2,0,0,20,2,22,4,24,1,36,5,38,7,54,1,60,3,70,1,72,3,80,1,82,688  
399 DATA 7,94,5,96,1,98,4,100,2,0,0,20,2,22,4,24,1,26,8,36,5,38,7,74,1,675  
400 DATA 76,3,80,1,82,7,84,8,94,5,96,1,98,4,100,2,0,0,20,2,28,7,30,1,38,867  
401 DATA 7,42,1,44,3,48,1,52,3,56,1,58,3,80,1,88,7,90,1,98,7,100,2,0,0,793  
402 DATA 20,2,22,4,24,1,36,5,38,7,44,1,46,3,48,3,66,1,68,3,80,1,82,7,94,706  
403 DATA 5,96,1,98,4,100,2,0,0,20,2,22,4,24,1,26,8,36,5,38,7,80,1,82,7,669  
404 DATA 84,8,94,5,96,1,98,4,100,2,0,0,2,0,2,28,7,30,1,38,7,48,1,52,3,56,785  
405 DATA 1,58,8,60,1,62,3,80,1,88,7,90,1,98,7,100,2,0,0,20,2,24,1,26,3,743  
406 DATA 30,1,32,3,38,1,42,3,56,1,58,3,66,1,68,3,80,1,82,3,98,1,100,2,0,773  
407 DATA 0,20,2,22,3,60,1,62,3,94,1,96,3,98,1,100,2,0,0,20,2,30,1,32,3,656  
408 DATA 48,1,50,3,84,1,86,3,98,1,100,2,0,0,20,2,26,1,28,8,44,1,46,3,76,732  
409 DATA 1,78,3,90,1,94,3,98,1,100,2,0,0,20,2,42,1,44,3,64,1,66,3,70,1,788  
410 DATA 72,3,82,1,84,3,98,1,100,2,0,0,2,0,2,36,1,40,3,98,1,100,2,0,0,20,769  
411 DATA 2,32,1,34,3,56,1,58,8,64,3,88,1,94,3,98,1,100,2,0,0,20,2,70,1,742  
412 DATA 72,3,80,1,82,8,94,1,96,3,98,1,100,2,0,0,20,2,50,1,54,3,84,1,86,942  
413 DATA 3,98,1,100,2,0,0,100,2,0,0,100,2,0,0,20,2,28,1,30,3,46,1,50,3,592  
414 DATA 66,1,68,3,78,1,82,3,90,1,92,3,98,1,100,2,0,0,20,2,24,1,26,3,34,799  
415 DATA 1,36,3,52,1,54,3,72,1,74,3,98,1,100,2,0,0,20,2,98,1,100,2,0,0,724  
416 DATA 20,2,40,7,48,1,52,3,60,1,64,3,78,1,98,7,100,2,0,0,20,2,26,5,28,668  
417 DATA 8,40,1,44,3,52,1,54,3,72,1,76,3,86,1,90,5,98,1,100,2,0,0,20,2,763  
418 DATA 54,1,56,3,58,1,60,3,84,1,88,5,98,8,98,1,100,2,0,0,20,2,28,1,34,798  
419 DATA 3,46,1,50,3,54,1,56,3,66,1,70,3,80,1,84,3,92,1,94,3,98,1,100,2,916  
420 DATA 0,0,20,2,50,1,52,8,54,5,98,1,100,2,0,0,20,2,98,1,100,2,0,0,20,636

421 DATA 2,40,7,50,1,54,3,78,1,98,7,100,2,0,0,20,2,36,1,36,3,42,1,44,3,631  
422 DATA 58,1,62,3,72,1,74,3,76,8,88,1,92,3,98,1,100,2,0,0,20,2,22,3,38,828  
423 DATA 1,40,3,80,1,84,3,98,1,100,2,0,0,20,2,46,1,48,3,98,1,100,2,0,0,734  
424 DATA 20,2,66,1,68,3,98,1,100,2,0,0,2,0,2,38,1,40,3,70,1,72,3,98,1,100,810  
425 DATA 2,0,0,20,2,22,3,26,1,34,5,44,1,48,3,58,1,64,3,86,1,88,3,98,1,614  
426 DATA 100,2,0,0,20,2,26,1,28,8,40,1,42,3,52,1,56,3,90,1,92,3,98,1,100,770  
427 DATA 2,0,0,20,2,26,1,28,5,30,1,34,3,54,1,58,3,64,1,66,3,72,1,74,3,552  
428 DATA 88,1,90,3,92,8,98,1,100,2,0,0,2,0,2,26,1,28,3,36,1,40,3,52,1,54,750  
429 DATA 3,66,1,70,3,84,1,88,3,98,1,100,2,0,0,20,2,30,1,32,3,94,1,96,3,802  
430 DATA 98,1,100,2,0,0,20,2,60,1,62,6,98,1,100,2,0,0,100,2,0,0,100,2,0,757  
431 DATA 0,20,2,98,1,100,2,0,0,20,2,98,1,100,2,0,0,20,2,98,1,100,2,0,0,669  
432 DATA 20,2,98,1,100,2,0,0,20,2,34,1,58,7,60,1,84,7,98,1,100,2,0,0,20,718  
433 DATA 2,34,1,36,7,38,4,48,8,58,5,60,1,70,5,80,8,82,4,84,7,98,1,100,2,843  
434 DATA 0,0,20,2,34,1,36,7,38,4,58,7,60,1,80,7,82,4,84,7,98,1,100,2,0,733  
435 DATA 0,20,2,34,1,36,7,38,4,40,7,42,4,44,8,46,5,48,8,50,5,52,8,58,5,572  
436 DATA 60,1,66,5,68,8,70,5,72,8,74,5,76,8,78,4,80,7,82,4,84,7,98,1,100,1071  
437 DATA 2,0,0,20,2,34,1,36,7,38,4,40,7,42,4,76,7,78,4,80,7,82,4,84,7,666  
438 DATA 98,1,100,2,0,0,20,2,34,1,36,7,38,4,40,7,42,4,44,7,74,5,76,7,78,727  
439 DATA 4,80,7,82,4,84,7,98,1,100,2,0,0,20,2,34,1,36,7,38,4,40,7,42,4,76,7,78,4,80,7,82,4,676  
440 DATA 44,1,74,5,76,1,78,4,80,7,82,4,84,7,98,1,100,2,0,0,20,2,34,1,36,841  
441 DATA 7,38,4,40,7,42,4,44,7,46,8,48,5,50,8,52,5,54,8,56,5,58,8,60,5,669  
442 DATA 62,8,64,5,66,8,68,5,70,8,72,5,74,8,76,7,78,4,80,7,82,4,84,7,98,1050  
443 DATA 1,100,2,0,0,20,2,34,1,36,7,38,4,40,7,42,4,76,7,78,4,80,7,82,4,676  
444 DATA 84,7,98,1,100,2,0,0,20,2,34,1,36,7,38,4,40,7,42,4,44,8,46,5,48,678  
445 DATA 8,50,5,52,8,58,5,60,1,66,5,68,8,70,5,72,8,74,5,76,8,78,4,80,7,881  
446 DATA 82,4,84,7,98,1,100,2,0,0,20,2,34,1,36,7,38,4,58,7,60,1,80,7,82,815  
447 DATA 4,84,7,98,1,100,2,0,0,20,2,34,1,36,7,38,4,48,8,58,5,60,1,70,5,693  
448 DATA 80,8,82,4,84,7,98,1,100,2,0,0,2,0,2,34,1,58,7,60,1,84,7,98,1,100,939  
449 DATA 2,0,0,20,2,98,1,100,2,0,0,20,2,98,1,100,2,0,0,20,2,98,1,100,2,0,0,20,2,671  
450 DATA 0,0,20,2,98,1,100,2,0,0,100,2,0,0,100,2,0,0,20,2,24,1,26,4,48,552  
451 DATA 1,50,7,54,3,56,7,60,3,62,7,66,3,68,7,98,1,100,2,0,0,20,2,32,1,710  
452 DATA 34,3,40,1,42,3,48,1,50,7,54,1,5





6,7,58,4,60,1,62,7,66,1,68,7,72,753  
453 DATA 1,74,3,84,1,86,5,98,1,100,2,0,0  
,20,2,34,1,36,7,40,1,42,3,44,7,692  
454 DATA 48,1,50,7,52,1,54,4,56,7,60,1,6  
2,7,66,1,68,7,84,1,88,5,98,1,100,929  
455 DATA 2,0,0,20,2,24,1,38,7,40,1,46,7,  
48,1,50,7,54,1,56,7,58,1,60,4,535  
456 DATA 62,7,66,1,68,7,84,1,86,5,98,1,1  
00,2,0,0,20,2,34,1,36,7,42,1,44,775  
457 DATA 7,48,1,50,7,54,1,56,7,60,1,62,7,  
,64,4,66,1,68,7,84,1,86,8,90,5,845  
458 DATA 98,1,100,2,0,0,20,2,26,1,28,4,3  
4,1,36,7,38,3,42,1,44,7,48,1,50,594  
459 DATA 7,54,1,56,7,60,1,62,7,66,1,68,7,  
,98,1,100,2,0,0,20,2,34,1,36,7,698  
460 DATA 42,1,44,7,48,1,50,7,52,4,54,1,5  
6,7,60,1,62,7,66,1,68,7,90,1,92,829  
461 DATA 4,98,1,100,2,0,0,20,2,24,1,26,3  
,34,1,36,7,42,1,44,7,48,1,50,7,559  
462 DATA 54,1,56,7,60,1,62,7,64,4,66,1,6  
8,7,98,1,100,2,0,0,20,2,28,1,30,740  
463 DATA 4,34,1,36,7,38,1,40,3,42,4,44,7,  
,48,1,70,7,78,1,80,3,98,1,100,2,750  
464 DATA 0,0,20,2,34,1,44,7,98,1,100,2,0  
,0,20,2,34,1,36,7,42,1,44,7,84,587  
465 DATA 1,86,4,98,1,100,2,0,0,20,2,34,1  
,36,7,40,1,42,4,46,7,98,1,100,2,733  
466 DATA 0,0,20,2,24,1,36,7,42,1,44,7,56  
,1,58,3,68,1,74,5,98,1,100,2,0,651  
467 DATA 0,20,2,32,1,34,3,36,7,38,1,40,2  
,42,1,44,7,72,1,74,8,98,1,100,2,666  
468 DATA 0,0,20,2,30,1,32,4,34,1,36,7,42  
,1,44,7,48,1,50,8,58,1,60,3,72,562  
469 DATA 1,74,5,98,1,100,2,0,0,20,2,24,1  
,26,8,34,1,36,7,42,1,44,7,46,1,581  
470 DATA 50,3,60,1,62,8,82,1,90,5,92,8,9  
8,1,100,2,0,0,20,2,26,1,28,4,34,778  
471 DATA 1,36,7,38,4,42,1,44,7,52,1,54,4  
,98,1,100,2,0,0,20,2,34,1,36,7,592  
472 DATA 38,1,40,4,44,7,64,1,68,5,98,1,1  
00,2,0,0,20,2,22,5,24,8,26,5,32,617  
473 DATA 1,34,4,36,7,42,1,44,7,64,1,66,8  
,68,5,98,1,100,2,0,0,20,2,34,1,646  
474 DATA 36,7,38,1,40,4,42,1,44,7,52,1,5  
4,8,76,1,78,3,88,1,90,8,98,1,100,879  
475 DATA 2,0,0,20,2,34,1,36,7,42,1,44,7,  
98,1,100,2,0,0,20,2,30,0,100,2,601  
476 DATA 0,0,20,2,98,5,100,2,0,0,20,2,22  
,5,28,1,30,4,38,1,40,4,48,1,50,521  
477 DATA 4,58,1,60,4,68,1,70,4,78,1,80,4  
,88,1,90,4,96,1,98,5,100,2,0,0,918  
478 DATA 20,2,22,5,34,1,36,3,44,1,46,4,6  
4,1,66,3,76,1,78,3,86,1,88,3,96,784  
479 DATA 1,98,5,100,2,0,0,20,2,22,5,26,1  
,30,3,38,1,40,3,54,1,58,3,60,4,577  
480 DATA 64,1,66,3,84,1,88,3,96,1,98,5,1  
00,2,0,0,20,2,22,5,40,1,44,3,48,797  
481 DATA 4,50,1,56,3,58,4,72,1,74,3,88,1  
,90,3,96,1,98,5,100,2,0,0,20,2,832  
482 DATA 22,5,28,1,30,3,38,1,40,3,50,1,5  
2,3,58,1,62,3,74,1,76,3,86,1,90,732  
483 DATA 3,96,1,98,5,100,2,0,0,20,2,22,5  
,34,1,38,3,48,1,50,3,66,1,68,3,670  
484 DATA 76,1,80,3,84,4,96,1,98,5,100,2,

0,0,20,2,22,5,30,1,32,4,36,1,40,743  
485 DATA 3,56,1,58,3,70,1,72,4,84,1,86,3  
,96,1,98,5,100,2,0,0,20,2,22,5,793  
486 DATA 34,1,36,4,40,1,44,3,56,1,62,3,7  
4,1,78,3,84,1,86,3,96,1,98,5,100,915  
487 DATA 2,0,0,20,2,22,5,28,1,32,3,54,1,  
56,3,68,1,72,3,76,4,86,1,90,3,633  
488 DATA 96,1,98,5,100,2,0,0,20,2,22,5,5  
0,1,54,3,60,1,62,3,96,1,98,5,100,885  
489 DATA 2,0,0,20,2,22,5,28,1,44,3,46,4,  
54,1,56,3,66,1,68,4,70,1,84,3,588  
490 DATA 88,1,92,3,96,1,98,5,100,2,0,0,2  
0,2,22,5,96,1,98,5,100,2,0,0,20,857  
491 DATA 2,56,5,62,1,98,5,100,2,0,0,20,2  
,56,5,62,1,98,5,100,2,0,0,20,2,704  
492 DATA 56,5,62,1,98,5,100,2,0,0,20,2,5  
6,5,62,1,98,5,100,2,0,0,20,2,56,758  
493 DATA 5,62,1,98,5,100,2,0,0,20,2,56,5  
,62,1,98,5,100,2,0,0,20,2,56,5,707  
494 DATA 58,1,60,2,62,1,98,5,100,2,0,0,2  
0,2,30,5,56,8,62,1,88,8,98,5,100,872  
495 DATA 2,0,0,100,2,0,0,100,2,0,0,20,2,  
52,3,58,1,60,3,64,7,66,3,72,7,624  
496 DATA 98,3,100,2,0,0,20,2,22,3,24,1,3  
6,3,52,1,58,7,66,1,68,7,72,1,98,745  
497 DATA 3,100,2,0,0,20,2,22,3,64,1,66,5  
,68,7,72,1,74,7,98,1,100,2,0,0,718  
498 DATA 20,2,42,7,44,3,54,7,56,1,64,7,6  
6,5,68,7,98,1,100,2,0,0,20,2,48,724  
499 DATA 1,50,3,52,7,56,5,58,7,62,9,64,7  
,66,5,68,7,72,1,96,5,98,1,100,2,902  
500 DATA 0,0,20,2,22,1,24,7,26,1,48,5,50  
,1,54,7,56,5,60,7,62,3,64,7,66,598  
501 DATA 5,68,7,72,1,96,5,98,1,100,2,0,0  
,20,2,22,1,24,7,26,1,46,7,48,5,664  
502 DATA 50,7,52,3,54,7,56,5,60,7,62,1,6  
4,7,66,5,68,7,72,1,74,5,94,1,96,924  
503 DATA 5,98,1,100,2,0,0,20,2,22,1,24,7  
,26,1,28,7,50,5,52,1,58,5,60,7,582  
504 DATA 66,5,68,7,72,1,74,5,76,1,92,8,9  
4,1,96,5,98,1,100,2,0,0,20,2,22,916  
505 DATA 1,24,7,26,1,28,7,30,5,56,7,58,5  
,60,7,62,1,68,7,72,1,74,5,94,1,707  
506 DATA 96,5,98,1,100,2,0,0,20,2,22,1,2  
4,7,26,1,28,7,30,5,32,7,54,4,56,628  
507 DATA 7,58,5,64,7,72,1,74,5,76,9,78,5  
,80,9,82,5,84,9,86,5,88,9,90,5,1013  
508 DATA 92,9,94,5,96,9,98,1,100,2,0,0,2  
0,2,22,1,24,7,26,1,28,7,30,5,32,711  
509 DATA 7,54,1,56,7,58,5,64,7,98,1,100,  
2,0,0,20,2,22,1,24,7,26,1,28,7,598  
510 DATA 30,5,32,7,54,1,56,7,62,5,64,7,9  
8,1,100,2,0,0,20,2,26,1,28,7,30,645  
511 DATA 5,32,7,34,1,44,5,46,8,48,5,50,8  
,54,1,60,7,62,5,64,7,98,1,100,2,754  
512 DATA 0,0,20,2,22,7,24,1,26,3,28,7,30  
,5,32,7,34,1,36,8,48,1,50,5,54,451  
513 DATA 1,60,7,62,5,64,7,98,1,100,2,0,0  
,20,2,24,1,28,7,30,5,32,7,34,1,598  
514 DATA 36,5,48,1,50,5,56,1,60,7,62,5,6  
4,7,98,1,100,2,0,0,20,2,24,1,26,681  
515 DATA 7,28,3,30,5,32,7,34,1,36,8,48,1  
,50,5,56,1,60,7,62,5,64,7,98,1,656  
516 DATA 100,2,0,0,20,2,22,7,24,5,26,7,2



# ATARI



8,1,30,5,32,7,34,1,36,5,38,8,40,480  
 517 DATA 5,42,8,44,5,46,8,48,5,50,8,62,1,  
 64,7,66,3,98,1,100,2,0,0,20,2,695  
 518 DATA 22,7,24,5,26,7,30,5,32,7,62,1,6  
 4,7,66,3,86,7,98,1,100,2,0,0,20,682  
 519 DATA 2,22,7,24,9,28,7,30,5,32,7,56,1,  
 58,7,98,1,100,2,0,0,20,2,22,7,547  
 520 DATA 24,8,28,7,30,5,58,7,94,1,98,1,1  
 00,2,0,0,20,2,22,7,24,8,28,7,52,633  
 521 DATA 5,62,1,66,7,96,1,98,6,100,2,0,0  
 ,100,2,0,0,100,2,0,0,20,2,22,5,697  
 522 DATA 58,1,60,2,96,1,98,5,100,2,0,0,2  
 0,2,22,5,96,1,98,5,100,2,0,0,20,794  
 523 DATA 2,22,5,96,1,98,5,100,2,0,0,20,2  
 ,22,9,72,1,74,3,96,1,98,9,100,2,840  
 524 DATA 0,0,20,2,22,9,96,1,98,9,100,2,0  
 ,0,20,2,22,9,50,1,68,7,96,1,98,733  
 525 DATA 9,100,2,0,0,20,2,22,9,50,1,52,7  
 ,54,5,56,3,62,1,64,3,66,5,68,7,668  
 526 DATA 96,1,98,9,100,2,0,0,20,2,22,9,5  
 0,1,52,7,54,8,64,7,66,8,68,7,96,847  
 527 DATA 1,98,9,100,2,0,0,20,2,22,9,50,1  
 ,52,7,54,8,56,7,62,1,64,7,66,8,706  
 528 DATA 68,7,96,1,98,9,100,2,0,0,20,2,2  
 2,9,50,1,52,7,54,8,56,7,58,1,60,788  
 529 DATA 6,62,1,64,7,66,8,68,7,96,1,98,9  
 ,100,2,0,0,20,2,22,9,50,1,52,7,758  
 530 DATA 54,8,56,7,62,1,64,7,66,8,68,7,7  
 6,1,80,5,82,8,96,1,98,9,100,2,0,966  
 531 DATA 0,20,2,22,9,50,1,52,7,54,8,64,7  
 ,66,8,68,7,96,1,98,9,100,2,0,0,751  
 532 DATA 20,2,22,9,50,1,52,7,54,1,56,3,6  
 2,1,64,3,66,1,68,7,96,1,98,9,100,853  
 533 DATA 2,0,0,20,2,22,9,50,1,68,7,96,1,  
 98,9,100,2,0,0,20,2,22,9,96,1,637  
 534 DATA 98,9,100,2,0,0,20,2,22,9,96,1,9  
 8,9,100,2,0,0,20,2,22,9,96,1,98,816  
 535 DATA 9,100,2,0,0,20,2,22,9,96,1,98,9  
 ,100,2,0,0,20,2,22,9,96,1,98,9,727  
 536 DATA 100,2,0,0,20,2,22,9,96,1,98,9,1  
 00,2,0,0,20,2,22,9,96,1,98,9,100,818  
 537 DATA 2,0,0,100,2,0,0,100,2,0,0,20,2,  
 32,7,58,1,60,2,98,1,100,2,0,0,589  
 538 DATA 20,2,22,9,30,5,98,1,100,2,0,0,2  
 0,2,32,7,98,1,100,2,0,0,20,2,22,595  
 539 DATA 9,30,5,98,1,100,2,0,0,20,2,32,7  
 ,98,1,100,2,0,0,20,2,22,9,30,5,595  
 540 DATA 40,1,98,8,100,2,0,0,20,2,32,7,4  
 0,1,98,3,100,2,0,0,20,2,22,9,30,637  
 541 DATA 5,98,1,100,2,0,0,20,2,32,7,98,1  
 ,100,2,0,0,20,2,22,9,30,5,98,1,655  
 542 DATA 100,2,0,0,20,2,32,7,98,1,100,2,  
 0,0,20,2,22,9,30,5,98,1,100,2,0,653  
 543 DATA 0,20,2,32,7,98,1,100,2,0,0,20,2  
 ,22,9,30,5,98,1,100,2,0,0,20,2,573  
 544 DATA 32,7,98,1,100,2,0,0,20,2,22,9,3  
 0,5,98,1,100,2,0,0,20,2,32,7,98,688  
 545 DATA 1,100,2,0,0,20,2,22,9,30,5,98,1  
 ,100,2,0,0,20,2,32,7,98,1,100,2,654  
 546 DATA 0,0,20,2,22,9,30,5,98,1,100,2,0  
 ,0,20,2,32,7,98,1,100,2,0,0,100,651  
 547 DATA 2,0,0,100,2,0,0,20,2,38,1,40,3,  
 46,1,48,3,56,1,58,3,98,1,100,2,625  
 548 DATA 0,0,20,2,44,1,46,3,76,1,78,3,92

,1,94,3,98,1,100,2,0,0,20,2,26,713  
 549 DATA 1,28,3,38,1,40,3,54,1,56,3,60,1  
 ,62,3,84,1,86,3,98,1,100,2,0,0,729  
 550 DATA 20,2,28,1,30,3,48,1,50,3,68,1,7  
 0,3,88,1,90,3,94,1,96,3,98,1,100,903  
 551 DATA 2,0,0,20,2,34,1,36,3,54,1,58,3,  
 68,1,70,3,94,1,96,3,98,1,100,2,751  
 552 DATA 0,0,20,2,22,3,36,1,38,3,44,1,46  
 ,3,68,1,70,3,74,1,76,3,84,1,86,686  
 553 DATA 3,92,1,94,3,98,1,100,2,0,0,20,2  
 ,48,3,64,1,66,3,98,1,100,2,0,0,802  
 554 DATA 20,2,48,3,98,1,100,2,0,0,20,2,5  
 2,1,54,3,64,1,66,3,88,1,90,3,98,820  
 555 DATA 1,100,2,0,0,20,2,48,8,58,1,60,3  
 ,74,1,76,3,80,1,82,3,98,1,100,2,824  
 556 DATA 0,0,20,2,48,7,54,1,56,3,66,1,68  
 ,3,98,1,100,2,0,0,20,2,22,5,24,603  
 557 DATA 7,26,5,28,7,30,5,32,7,34,5,36,7  
 ,38,5,40,7,42,5,44,7,46,5,48,7,523  
 558 DATA 52,1,54,3,66,1,68,3,76,1,78,8,9  
 8,1,100,2,0,0,20,2,22,9,24,7,26,722  
 559 DATA 9,28,7,30,9,32,7,34,9,36,7,38,9  
 ,40,7,42,9,44,7,46,9,48,7,58,1,573  
 560 DATA 60,3,88,1,90,3,92,8,98,1,100,2,  
 0,0,20,2,22,5,24,7,26,5,28,7,30,722  
 561 DATA 5,32,7,34,5,36,7,38,5,40,7,42,5  
 ,44,7,46,5,48,7,52,1,54,3,82,1,613  
 562 DATA 86,1,88,3,94,1,96,3,98,1,100,2,  
 0,0,20,2,22,5,24,7,26,5,28,7,30,749  
 563 DATA 5,32,7,34,5,36,7,38,5,40,7,42,5  
 ,44,7,46,5,48,7,52,1,54,3,82,1,613  
 564 DATA 84,3,98,1,100,2,0,0,20,2,22,5,2  
 4,7,26,5,28,7,30,5,32,7,34,5,36,583  
 565 DATA 7,38,5,40,7,42,5,44,7,46,5,48,7  
 ,64,1,66,3,92,1,94,3,98,1,100,2,826  
 566 DATA 0,0,20,2,22,5,24,7,26,5,28,7,30  
 ,5,32,7,34,5,36,7,38,5,40,7,42,434  
 567 DATA 5,44,7,46,5,48,7,56,1,58,3,64,1  
 ,66,3,74,1,76,3,98,1,100,2,0,0,769  
 568 DATA 20,2,22,5,24,7,26,5,28,7,30,5,3  
 2,7,34,5,36,7,38,5,40,7,42,5,44,483  
 569 DATA 7,46,5,48,7,56,1,58,3,70,1,72,3  
 ,74,8,98,1,100,2,0,0,20,2,22,5,709  
 570 DATA 24,7,26,5,28,7,30,5,32,7,34,5,3  
 6,7,38,5,40,7,42,5,44,7,46,5,48,540  
 571 DATA 7,68,1,72,3,98,1,100,2,0,0,20,2  
 ,22,5,24,7,26,5,28,7,30,5,32,7,572  
 572 DATA 34,5,36,7,38,5,40,7,42,5,44,7,4  
 6,5,48,7,54,8,68,1,70,3,98,1,100,779  
 573 DATA 2,0,0,20,2,22,5,24,7,26,5,28,7,  
 30,5,32,7,34,5,36,7,38,5,40,7,394  
 574 DATA 42,5,44,7,46,5,48,7,56,1,58,3,9  
 0,1,92,3,98,1,100,2,0,0,100,2,0,811  
 575 DATA 0,100,2,0,0,20,2,24,1,26,3,64,1  
 ,66,3,96,1,100,2,0,0,20,2,36,1,570  
 576 DATA 38,3,54,1,56,3,98,1,100,2,0,0,2  
 0,2,32,1,34,3,92,1,94,3,98,1,100,837  
 577 DATA 2,0,0,20,2,50,1,52,3,66,1,68,8,  
 84,1,86,3,98,1,100,2,0,0,20,2,670  
 578 DATA 24,1,26,3,66,1,68,8,74,1,76,3,9  
 8,1,100,2,0,0,20,2,44,1,46,3,66,734  
 579 DATA 1,68,8,88,1,90,3,98,1,100,2,0,0  
 ,20,2,28,1,30,3,42,1,44,3,66,1,701  
 580 DATA 68,8,84,1,86,3,98,1,100,2,0,0,2



# ATARI



0,2,58,1,60,3,66,1,68,8,98,1,100,937  
 581 DATA 2,0,0,20,2,30,1,46,7,52,1,54,3,  
 58,1,60,3,66,1,68,8,70,1,86,7,647  
 582 DATA 98,1,100,2,0,0,20,2,24,1,28,3,3  
 0,1,32,7,44,4,46,7,66,1,68,8,70,663  
 583 DATA 1,72,7,84,4,86,7,90,1,92,3,98,1  
 ,100,2,0,0,20,2,30,1,32,7,34,4,778  
 584 DATA 42,1,44,4,46,7,54,1,58,3,66,1,6  
 8,8,70,1,72,7,74,4,82,1,84,4,86,888  
 585 DATA 7,98,1,100,2,0,0,20,2,30,1,32,7  
 ,34,4,42,1,44,4,46,7,52,1,54,3,592  
 586 DATA 64,1,66,3,70,1,72,7,74,4,82,1,8  
 4,4,86,7,98,1,100,2,0,0,20,2,30,879  
 587 DATA 1,32,7,34,4,42,1,44,4,46,7,56,1  
 ,58,3,70,1,72,7,74,4,82,1,84,4,739  
 588 DATA 86,7,98,1,100,2,0,0,20,2,30,1,3  
 2,7,34,4,42,1,44,4,46,7,70,1,72,711  
 589 DATA 7,74,4,82,1,84,4,86,7,88,3,94,1  
 ,96,3,98,1,100,2,0,0,20,2,24,1,882  
 590 DATA 26,3,30,1,32,7,44,4,46,7,50,1,5  
 2,3,70,1,72,7,84,4,86,7,90,1,92,820  
 591 DATA 3,98,1,100,2,0,0,20,2,30,1,46,7  
 ,60,1,62,3,70,1,86,7,98,1,100,2,801  
 592 DATA 0,0,20,2,30,1,32,3,46,1,48,3,54  
 ,1,56,3,70,1,72,3,86,1,88,3,98,722  
 593 DATA 1,100,2,0,0,20,2,40,1,42,3,90,1  
 ,92,3,98,1,100,2,0,0,20,2,26,1,647  
 594 DATA 28,3,30,8,58,1,60,3,98,1,100,2,  
 0,0,20,2,88,1,90,3,98,1,100,2,0,797  
 595 DATA 0,20,2,30,1,32,3,98,1,100,2,0,0  
 ,100,2,0,0,0,0,0,0,0,0,0,0,391  
 596 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
 597 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
 ,0,0,0,0,0,0,102,102,102,306  
 598 DATA 0,0,0,0,0,102,255,102,102,255,1  
 02,0,24,62,96,60,6,124,24,0,0,102,108,24  
 ,48,1596  
 599 DATA 102,70,0,28,54,28,56,111,102,59  
 ,0,0,24,24,24,0,0,0,0,0,14,28,24,24,28,8  
 00  
 600 DATA 14,0,0,112,56,24,24,56,112,0,0,  
 102,60,255,60,102,0,0,0,24,24,126,24,24,  
 0,1199  
 601 DATA 0,0,0,0,0,0,24,24,48,0,0,0,126,  
 0,0,0,0,0,0,0,0,24,24,0,270  
 602 DATA 0,6,12,24,48,96,64,0,0,60,102,1  
 10,118,102,60,0,0,24,56,24,24,24,126,0,0  
 ,1080  
 603 DATA 60,102,12,24,48,126,0,0,126,12,  
 24,12,102,60,0,0,12,28,60,108,126,12,0,0  
 ,126,1180  
 604 DATA 96,124,6,102,60,0,0,60,96,124,1  
 02,102,60,0,0,126,6,12,24,48,48,0,0,60,1  
 02,1358  
 605 DATA 60,102,102,60,0,0,60,102,62,6,1  
 2,56,0,0,0,24,24,0,24,24,0,0,0,24,24,766  
 606 DATA 0,24,24,48,6,12,24,48,24,12,6,0  
 ,0,0,126,0,0,126,0,0,96,48,24,12,24,684  
 607 DATA 48,96,0,0,60,102,12,24,0,24,0,0  
 ,60,102,110,110,96,62,0,0,24,60,102,102,  
 126,1320  
 608 DATA 102,0,0,124,102,124,102,102,124  
 ,0,0,60,102,96,96,102,60,0,0,120,108,102

,102,108,120,1956  
 609 DATA 0,0,126,96,124,96,96,126,0,0,12  
 6,96,124,96,96,96,0,0,62,96,96,110,102,6  
 2,0,1826  
 610 DATA 0,102,102,126,102,102,102,0,0,1  
 26,24,24,24,24,126,0,0,6,6,6,6,102,60,0,  
 0,1170  
 611 DATA 102,108,120,120,108,102,0,0,96,  
 96,96,96,96,126,0,0,99,119,127,107,99,99  
 ,0,0,102,2018  
 612 DATA 118,126,126,110,102,0,0,60,102,  
 102,102,102,60,0,0,124,102,102,124,96,96  
 ,0,0,60,102,1916  
 613 DATA 102,102,108,54,0,0,124,102,102,  
 124,108,102,0,0,60,96,60,6,6,60,0,0,126,  
 24,24,1490  
 614 DATA 24,24,24,0,0,102,102,102,102,10  
 2,126,0,0,102,102,102,102,60,24,0,0,99,9  
 9,107,127,1632  
 615 DATA 119,99,0,0,102,102,60,60,102,10  
 2,0,0,102,102,60,24,24,24,0,0,126,12,24,  
 48,96,1388  
 616 DATA 126,0,0,30,24,24,24,24,30,0,0,6  
 4,96,48,24,12,6,0,0,120,24,24,24,24,120,  
 868  
 617 DATA 0,0,8,28,54,99,0,0,0,0,0,0,0,0,  
 0,255,0,0,54,127,127,62,28,8,0,850  
 618 DATA 12,15,51,51,51,15,3,15,48,240,2  
 04,204,204,240,192,240,49,19,1,3,13,12,1  
 2,20,76,1990  
 619 DATA 196,64,192,112,48,48,20,3,15,51  
 ,51,51,15,3,1,192,240,240,240,240,240,19  
 2,64,7,1,2526  
 620 DATA 3,13,48,48,48,80,192,64,192,64,  
 252,1,1,1,3,15,15,15,15,15,3,1,192,240,2  
 04,1725  
 621 DATA 204,204,240,192,64,3,1,3,1,63,6  
 4,64,64,208,64,192,112,12,12,12,5,255,19  
 5,195,243,2672  
 622 DATA 243,227,227,255,162,41,138,162,  
 74,162,42,166,162,138,40,161,170,104,42,  
 136,162,40,106,10,168,3338  
 623 DATA 170,26,136,166,170,138,164,162,  
 106,34,170,5,17,68,211,252,243,207,206,6  
 4,16,84,21,213,197,3246  
 624 DATA 253,255,240,207,203,242,60,63,1  
 2,3,143,207,227,252,188,252,240,192,21,3  
 7,42,0,81,82,162,3664  
 625 DATA 0,255,195,195,243,243,227,227,2  
 55,24,24,24,255,255,0,0,0,0,16,1,18,32,0  
 ,2,64,2555  
 626 DATA 0,68,0,36,8,168,0,136,0,72,34,8  
 ,72,1,16,0,32,8,32,136,132,64,0,0,0,1023  
 627 DATA 0,0,1,0,18,32,16,0,0,16,32,4,0,  
 16,136,0,16,32,2,4,4,0,0,16,32,377  
 628 DATA 160,4,128,64,0,0,2,3,10,15,37,6  
 3,170,255,128,192,160,240,88,252,170,255  
 ,85,191,42,2714  
 629 DATA 47,5,11,2,2,85,254,168,248,80,2  
 24,128,128,170,170,128,128,133,133,132,1  
 32,170,170,2,2,2852  
 630 DATA 82,82,18,18,132,132,133,133,128  
 ,128,170,170,18,18,82,82,2,2,170,170,240  
 ,252,255,63,15,2695





631 DATA 63,255,255,3,15,63,255,252,240,  
252,255,255,255,63,15,63,255,252,240,255  
,252,240,252,255,63,4623  
632 DATA 15,3,170,170,128,128,133,133,13  
2,132,170,170,128,128,133,133,132,132,17  
0,170,128,128,133,133,132,3264  
633 DATA 132,64,128,80,88,164,252,1,0,2,  
1,5,53,42,63,0,192,253,0,84,88,144,240,1  
28,192,2396  
634 DATA 63,192,53,53,10,15,2,3,64,128,8  
0,88,164,252,1,0,64,128,80,88,164,252,1,  
0,64,2009  
635 DATA 128,80,88,164,252,1,0,0,0,0,0,0  
,4,16,1,0,0,0,0,16,0,64,4,0,4,822  
636 DATA 16,1,0,0,0,0,64,0,16,64,0,0,0,0  
,0,0,0,1,0,18,32,16,0,0,16,244  
637 DATA 32,4,0,16,136,0,16,32,2,4,4,0,0  
,16,32,160,4,128,64,0,0,16,24,28,30,748  
638 DATA 26,24,16,0,76,35,124,112,112,11  
2,68,203,124,4,4,4,4,4,4,4,4,4,4,108  
2  
639 DATA 4,112,6,6,112,6,112,112,6,112,1  
12,65,3,73,169,3,141,48,2,169,124,141,49  
,2,169,1858  
640 DATA 120,141,244,2,169,3,141,21,156,  
169,0,141,7,210,141,200,2,173,78,127,141  
,180,144,169,8,2887  
641 DATA 141,196,2,169,36,141,197,2,169,  
84,141,198,2,169,120,141,199,2,169,0,141  
,13,156,173,76,2837  
642 DATA 127,141,47,127,141,187,144,162,  
19,189,149,144,157,51,127,173,70,127,157  
,149,144,202,224,13,208,3379  
643 DATA 239,32,79,127,173,120,2,201,11,  
208,19,173,47,127,205,76,127,240,11,206,  
47,127,173,13,156,2939  
644 DATA 233,16,141,13,156,173,120,2,201  
,7,208,19,173,47,127,205,77,127,240,11,2  
38,47,127,173,13,2894  
645 DATA 156,105,16,141,13,156,169,0,133  
,77,162,255,160,255,136,208,253,202,208,  
248,173,47,127,141,187,3728  
646 DATA 144,173,132,2,208,179,96,77,77,  
77,77,77,77,77,77,77,77,77,77,77,77,77,  
7,77,77,2320  
647 DATA 77,77,77,77,77,77,77,77,77,77,7  
7,77,77,77,77,77,77,77,77,77,77,77,77,78  
,79,1928  
648 DATA 86,86,86,79,78,79,86,86,78,79,8  
6,79,78,86,78,86,78,79,78,79,86,86,86,79  
,78,2045  
649 DATA 86,86,86,86,79,86,86,86,79,78,7  
9,77,77,80,81,86,81,80,86,80,86,86,86,86  
,81,2070  
650 DATA 86,81,80,86,80,86,80,81,80,81,8  
6,80,86,86,81,86,80,81,80,81,86,80,81,86  
,81,2062  
651 DATA 80,77,77,78,79,86,86,86,79,78,8  
6,78,79,86,79,86,79,78,86,78,86,78,79,78  
,79,2021  
652 DATA 86,78,79,86,78,86,86,86,78,79,8  
6,86,86,79,78,79,77,77,80,81,86,81,80,86  
,80,2044  
653 DATA 86,86,86,86,81,86,86,86,86,80,8

6,80,81,80,81,86,81,86,86,80,86,80,81,80  
,81,2084  
654 DATA 86,81,86,81,80,81,77,77,78,79,8  
6,86,86,79,78,79,86,86,78,79,78,86,86,79  
,78,2036  
655 DATA 86,86,86,86,79,86,86,86,79,78,8  
6,86,86,86,79,86,79,78,86,78,79,77,77,80  
,81,2062  
656 DATA 80,81,80,81,80,81,80,81,80,81,8  
0,81,80,81,80,81,80,81,80,81,80,81,80,81  
,80,2012  
657 DATA 81,80,81,80,81,80,81,80,81,80,8  
1,77,77,78,79,78,79,78,79,78,79,78,79,86  
,86,1997  
658 DATA 86,79,78,79,86,86,78,79,78,86,8  
6,86,78,86,78,79,86,79,78,79,78,79,78,79  
,78,2022  
659 DATA 79,77,77,80,81,80,81,80,81,80,8  
1,80,81,86,81,86,86,80,86,86,86,86,81,86  
,81,2049  
660 DATA 80,81,80,86,80,81,86,81,80,81,8  
0,81,80,81,80,81,77,77,78,79,78,79,65,66  
,78,1976  
661 DATA 79,78,79,86,78,78,86,78,86,78,7  
8,86,78,78,86,86,78,78,86,86,86,86,79,78  
,79,2034  
662 DATA 78,79,78,79,78,79,77,77,80,81,8  
0,81,67,68,80,81,80,81,86,78,86,86,78,86  
,86,1990  
663 DATA 86,86,78,78,78,78,86,78,86,78,7  
8,86,81,80,81,80,81,80,81,80,81,77,77,78  
,79,2012  
664 DATA 78,79,78,79,78,79,78,79,86,86,8  
6,78,78,86,78,78,86,78,86,86,86,78,78,86  
,78,2026  
665 DATA 78,86,79,78,79,78,79,78,79,78,7  
9,77,77,80,81,80,81,80,81,80,81,80,81,80  
,81,1991  
666 DATA 80,81,80,81,80,81,80,81,80,81,8  
0,81,80,81,80,81,80,81,80,81,80,81,80,81  
,80,2012  
667 DATA 81,77,77,77,77,77,77,77,77,77,7  
7,77,77,77,77,77,77,77,77,77,77,77,77,77  
,77,1929  
668 DATA 77,77,77,77,77,77,77,77,77,77,7  
7,77,77,77,77,77,77,0,0,0,0,0,0,8,35,135  
2  
669 DATA 9,0,17,25,24,22,0,0,0,0,0,0,0,2  
46,239,238,128,239,236,233,246,229,242,1  
28,227,2728  
670 DATA 249,242,225,238,235,225,0,0,0,0  
,0,0,0,0,35,33,54,37,0,33,0,0,0,0,0,1606  
671 DATA 0,0,0,176,181,174,171,180,165,1  
28,154,128,144,144,144,144,144,144,0,0,0  
,0,0,33,48,2402  
672 DATA 19,169,0,162,128,157,0,41,202,2  
08,250,169,42,133,30,169,20,133,29,169,8  
7,162,0,129,29,2637  
673 DATA 230,29,165,29,201,102,208,242,1  
69,0,129,29,230,30,165,30,201,86,208,226  
,169,255,141,252,2,3328  
674 DATA 96,206,56,128,240,3,76,15,128,1  
69,2,141,56,128,172,23,128,162,6,189,16,  
128,157,17,128,2570





,191,133,141,192,133,141,193,133,32,38,1  
31,96,173,57,130,201,3208  
321 DATA 8,208,24,169,101,141,190,133,16  
9,102,141,191,133,169,104,141,192,133,16  
9,103,141,193,133,32,38,3258  
322 DATA 131,96,173,57,130,201,9,208,24,  
169,112,141,190,133,169,113,141,191,133,  
169,115,141,192,133,169,3440  
323 DATA 114,141,193,133,32,38,131,96,96  
,162,0,173,190,133,129,218,230,219,173,1  
91,133,129,218,230,219,3720  
324 DATA 173,192,133,129,218,198,218,173  
,193,133,129,218,96,204,97,90,142,145,15  
1,143,152,149,20,101,24,3621  
325 DATA 102,8,96,0,67,132,100,118,151,1  
49,145,145,149,148,20,46,102,30,96,0,0,9  
1,100,118,144,2257  
326 DATA 146,144,153,149,52,229,30,102,1  
4,96,0,194,133,100,118,153,153,145,144,1  
44,4,229,30,126,14,2802  
327 DATA 122,0,0,96,100,118,146,149,145,  
144,144,68,229,96,102,80,96,0,254,99,100  
,136,145,146,144,2859  
328 DATA 146,149,116,165,26,136,10,128,0  
,14,102,20,122,148,144,145,150,149,132,2  
29,60,102,44,96,0,2533  
329 DATA 194,103,60,118,146,149,145,145,  
149,4,53,60,138,40,128,0,174,105,40,126,  
145,152,145,145,149,2813  
330 DATA 100,69,64,122,44,112,0,40,108,6  
0,138,146,144,145,145,149,116,37,60,136,  
40,128,0,12,110,2225  
331 DATA 100,140,149,144,145,149,149,84,  
229,24,102,10,96,0,160,112,60,100,150,14  
4,145,145,144,52,229,2962  
332 DATA 60,102,40,96,0,38,114,60,100,14  
6,149,145,145,144,132,53,60,138,40,134,0  
,28,115,60,142,2241  
333 DATA 146,149,145,145,144,148,213,60,  
122,40,108,0,214,117,98,100,147,149,145,  
145,144,84,117,40,102,3022  
334 DATA 20,96,0,0,94,60,100,149,149,144  
,145,144,4,229,60,138,40,138,0,100,2,0,0  
,20,2,1834  
335 DATA 54,1,56,6,88,1,90,5,92,8,94,3,9  
6,7,98,1,100,2,0,0,20,2,94,1,96,1015  
336 DATA 7,98,1,100,2,0,0,20,2,26,1,32,3  
,36,1,38,3,46,1,48,3,58,1,64,3,594  
337 DATA 80,1,82,3,94,1,96,7,98,1,100,2,  
0,0,20,2,56,1,60,3,72,1,88,3,94,965  
338 DATA 1,96,7,98,1,100,2,0,0,20,2,94,1  
,96,7,98,1,100,2,0,0,20,2,40,1,789  
339 DATA 44,3,60,1,62,3,70,1,76,3,94,1,9  
6,7,98,1,100,2,0,0,20,2,70,1,74,889  
340 DATA 3,94,1,96,7,98,1,100,2,0,0,20,2  
,32,1,36,3,50,1,56,3,94,1,96,7,804  
341 DATA 98,1,100,2,0,0,20,2,26,1,30,3,4  
4,1,48,3,66,1,70,3,94,1,96,7,98,815  
342 DATA 1,100,2,0,0,20,2,94,1,96,7,98,1  
,100,2,0,0,20,2,40,1,44,3,60,1,695  
343 DATA 62,3,70,1,76,3,94,1,96,7,98,1,1  
00,2,0,0,20,2,94,1,96,7,98,1,100,1033  
344 DATA 2,0,0,20,2,94,1,96,7,98,1,100,2  
,0,0,20,2,24,1,26,3,30,1,36,3,569  
345 DATA 54,1,70,5,80,8,84,5,94,1,96,7,9  
8,1,100,2,0,0,20,2,94,1,96,7,98,1024  
346 DATA 1,100,2,0,0,20,2,94,1,96,7,98,1

,100,2,0,0,20,2,50,1,56,3,94,1,751  
347 DATA 96,7,98,1,100,2,0,0,20,2,26,1,8  
6,7,94,1,96,7,98,1,100,2,0,0,20,865  
348 DATA 2,26,1,86,9,94,1,96,7,98,1,100,  
2,0,0,20,2,26,1,86,3,94,1,96,7,859  
349 DATA 98,1,100,2,0,0,20,2,94,1,96,7,9  
8,1,100,2,0,0,100,2,0,0,96,0,0,820  
350 DATA 0,0,100,2,0,0,20,2,98,1,100,2,0  
,0,20,2,50,1,52,6,98,1,100,2,0,657  
351 DATA 0,20,2,30,1,36,3,50,1,52,3,70,1  
,74,3,98,1,100,2,0,0,20,2,98,1,668  
352 DATA 100,2,0,0,20,2,46,1,48,3,56,1,6  
0,3,82,1,84,3,98,1,100,2,0,0,20,733  
353 DATA 2,56,1,58,3,98,1,100,2,0,0,20,2  
,98,1,100,2,0,0,20,2,40,1,46,3,656  
354 DATA 70,1,72,3,98,1,100,2,0,0,20,2,9  
8,1,100,2,0,0,20,2,98,1,100,2,0,793  
355 DATA 0,20,2,98,1,100,2,0,0,20,2,26,5  
,98,9,100,2,0,0,20,2,98,1,100,2,708  
356 DATA 0,0,20,2,98,1,100,2,0,0,20,2,98  
,1,100,2,0,0,20,2,98,1,100,2,0,667  
357 DATA 0,20,2,98,1,100,2,0,0,20,2,98,1  
,100,2,0,0,20,2,98,1,100,2,0,0,667  
358 DATA 20,2,98,1,100,2,0,0,20,2,98,1,1  
00,2,0,0,100,2,0,0,169,0,141,28,156,1042  
359 DATA 141,1,210,141,3,210,173,11,156,  
133,218,173,12,156,133,219,169,88,162,0,  
129,218,230,218,129,3433  
360 DATA 218,230,219,129,218,198,218,129  
,218,169,12,162,255,160,255,136,208,253,  
202,208,248,233,1,208,242,4729  
361 DATA 169,149,141,6,144,169,144,141,7  
,144,169,0,141,26,156,169,255,141,252,2,  
96,3,96,0,0,2720  
362 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0  
,0,0,0,0,173,31,208,201,7,620  
363 DATA 240,5,104,104,76,0,144,169,84,1  
33,220,169,16,133,219,169,0,133,77,169,0  
,141,252,137,230,3124  
364 DATA 219,230,219,166,219,224,102,240  
,35,162,0,161,219,201,82,240,50,201,225,  
240,40,32,238,136,161,4042  
365 DATA 219,201,65,240,11,201,69,240,7,  
201,73,240,3,76,29,135,76,237,141,169,16  
,133,219,198,220,3419  
366 DATA 198,220,166,220,224,42,16,197,7  
6,162,139,32,177,136,76,100,135,32,146,1  
36,165,219,133,221,165,3533  
367 DATA 220,133,222,162,0,230,222,230,2  
22,161,221,208,9,32,153,135,32,174,135,7  
6,150,135,201,78,240,3781  
368 DATA 19,201,65,240,15,201,69,240,11,  
201,73,240,7,201,1,240,3,32,20,136,76,29  
,135,165,220,2840  
369 DATA 133,222,169,0,129,221,230,221,1  
29,221,230,222,129,221,198,221,129,221,9  
6,165,220,133,222,230,222,4534  
370 DATA 230,222,173,142,136,129,221,230  
,221,173,143,136,129,221,230,222,173,144  
,136,129,221,198,221,173,145,4498  
371 DATA 136,129,221,230,222,161,221,201  
,1,240,24,201,0,240,20,32,213,136,173,18  
7,142,141,101,147,173,3692  
372 DATA 186,142,141,93,147,169,0,141,23  
,156,230,221,161,221,201,102,208,13,198,  
221,169,1,141,181,141,3607  
373 DATA 32,24,141,76,19,136,201,113,208



# ATARI



,10,198,221,169,0,141,181,141,32,24,141,  
96,165,220,133,222,3044  
374 DATA 169,0,141,237,136,173,10,210,16  
,46,173,237,136,201,2,240,38,230,221,230  
,221,161,221,208,81,3738  
375 DATA 230,222,230,222,161,221,208,73,  
165,219,133,221,32,153,135,230,221,230,2  
21,165,220,133,222,32,182,4481  
376 DATA 135,230,219,230,219,96,173,237,  
136,201,2,240,42,165,220,133,222,165,219  
,133,221,198,221,198,221,4476  
377 DATA 161,221,208,33,230,222,230,222,  
161,221,208,25,165,219,133,221,32,153,13  
5,198,221,198,221,165,220,4423  
378 DATA 133,222,32,182,135,96,238,237,1  
36,76,80,136,238,237,136,76,34,136,0,0,0  
,0,169,82,141,2952  
379 DATA 142,136,169,83,141,143,136,169,  
85,141,144,136,169,84,141,145,136,169,80  
,141,186,142,169,0,141,3328  
380 DATA 187,142,96,169,225,141,142,136,  
169,226,141,143,136,169,228,141,144,136,  
169,227,141,145,136,173,10,3872  
381 DATA 210,41,10,105,10,141,186,142,16  
9,32,141,187,142,96,201,65,240,9,201,69,  
240,5,201,73,240,3156  
382 DATA 1,96,169,1,133,223,141,181,141,  
32,24,141,96,0,201,233,208,17,173,20,156  
,201,127,16,11,2742  
383 DATA 173,19,156,201,8,16,16,76,48,13  
7,96,32,28,137,32,39,137,32,227,137,76,3  
,137,32,28,2023  
384 DATA 137,32,39,137,32,253,137,76,3,1  
37,169,3,141,21,156,169,0,141,7,210,96,1  
65,219,133,221,2634  
385 DATA 165,220,133,222,96,32,39,137,19  
8,221,198,221,162,0,161,221,201,78,240,1  
0,201,0,240,6,32,3434  
386 DATA 205,141,76,79,137,32,190,137,76  
,3,137,32,39,137,230,221,230,221,162,0,1  
61,221,201,78,240,3386  
387 DATA 10,201,0,240,6,32,205,141,76,11  
4,137,32,190,137,230,219,230,219,76,3,13  
7,32,39,137,230,3073  
388 DATA 222,230,222,162,0,161,221,201,7  
8,240,10,201,0,240,6,32,205,141,76,145,1  
37,32,190,137,76,3365  
389 DATA 3,137,32,39,137,198,222,198,222  
,162,0,161,221,201,78,240,24,201,0,240,2  
0,32,205,141,238,3352  
390 DATA 252,137,173,252,137,205,20,156,  
208,3,238,19,156,76,3,137,32,190,137,76,  
3,137,173,10,210,3140  
391 DATA 201,10,16,29,201,7,48,25,169,23  
3,129,221,230,221,169,234,129,221,230,22  
2,169,236,129,221,198,3898  
392 DATA 221,169,235,129,221,238,20,156,  
96,169,82,162,0,129,221,230,221,169,83,1  
29,221,230,222,169,85,4007  
393 DATA 129,221,198,221,169,84,129,221,  
96,0,169,225,162,0,129,221,230,221,169,2  
26,129,221,230,222,169,4191  
394 DATA 228,129,221,198,221,169,227,129  
,221,96,201,100,48,69,162,0,201,101,208,

13,32,186,138,32,151,3481  
395 DATA 138,32,96,138,32,221,138,96,201  
,109,208,13,32,151,138,32,96,138,32,221,  
138,32,186,138,96,2652  
396 DATA 201,110,208,13,32,221,138,32,18  
6,138,32,151,138,32,96,138,96,201,111,20  
8,12,32,96,138,32,2792  
397 DATA 221,138,32,186,138,32,151,138,9  
6,32,39,137,230,221,230,221,161,221,240,  
7,201,1,240,3,76,3392  
398 DATA 130,138,169,237,141,39,140,32,2  
6,140,32,138,139,32,131,138,104,104,96,3  
2,39,137,169,0,129,2612  
399 DATA 221,230,221,129,221,230,222,129  
,221,198,221,129,221,96,32,39,137,198,22  
2,198,222,161,221,240,7,4366  
400 DATA 201,1,240,3,76,185,138,169,229,  
141,39,140,32,26,140,32,138,139,32,131,1  
38,104,104,96,32,2706  
401 DATA 39,137,198,221,198,221,161,221,  
240,7,201,1,240,3,76,220,138,169,238,141  
,39,140,32,26,140,3447  
402 DATA 32,138,139,32,131,138,104,104,9  
6,32,39,137,230,222,230,222,161,221,240,  
7,201,1,240,3,76,3176  
403 DATA 255,138,169,239,141,39,140,32,2  
6,140,32,138,139,32,131,138,104,104,96,1  
60,0,185,189,142,133,3042  
404 DATA 219,200,185,189,142,133,220,200  
,162,0,161,219,201,237,240,44,201,239,24  
0,40,201,238,240,36,201,4388  
405 DATA 229,240,32,201,240,240,28,201,2  
44,240,24,201,245,240,20,201,246,240,16,  
204,188,142,48,203,162,4275  
406 DATA 0,169,0,157,188,142,232,208,250  
,96,32,39,137,161,219,233,128,129,219,16  
5,219,133,221,165,220,3862  
407 DATA 133,222,198,222,198,222,161,221  
,201,82,240,7,201,225,240,3,76,135,139,1  
65,220,133,222,230,221,4317  
408 DATA 161,221,201,102,208,13,169,1,14  
1,181,141,198,221,32,24,141,76,135,139,1  
69,0,141,181,141,198,3335  
409 DATA 221,32,24,141,76,50,139,173,39,  
140,129,221,230,221,169,102,129,221,230,  
222,169,104,129,221,198,3730  
410 DATA 221,169,103,129,221,96,162,0,16  
9,0,157,188,142,232,208,250,160,0,169,86  
,133,220,169,16,133,3533  
411 DATA 219,230,219,230,219,166,219,224  
,102,240,70,162,0,161,219,201,78,240,238  
,201,0,240,234,201,123,4436  
412 DATA 208,3,76,47,142,201,89,208,3,76  
,87,142,201,93,208,3,76,98,142,201,119,2  
08,3,76,109,2819  
413 DATA 142,201,251,208,3,76,120,142,20  
1,217,208,3,76,131,142,201,221,208,3,76,  
142,142,32,22,138,3306  
414 DATA 32,40,140,76,182,139,169,16,133  
,219,198,220,198,220,166,220,224,42,16,1  
62,140,188,142,76,0,3358  
415 DATA 139,165,221,153,189,142,165,222  
,200,153,189,142,200,96,0,201,112,48,71,  
162,0,201,112,208,13,3504







,246,144,173,244,144,141,245,144,162,9,1  
73,243,144,157,3,144,3505  
464 DATA 173,245,144,157,4,144,238,245,1  
44,232,232,232,224,80,48,234,96,16,42,0,  
32,212,144,76,121,3515  
465 DATA 148,216,173,9,156,205,243,144,1  
6,31,173,243,144,201,17,48,24,238,118,14  
8,173,118,148,201,4,3339  
466 DATA 208,11,206,243,144,32,212,144,1  
69,0,141,118,148,141,4,212,173,9,156,205  
,243,144,48,31,173,3315  
467 DATA 243,144,201,59,16,24,206,118,14  
8,173,118,148,201,255,208,11,238,243,144  
,32,212,144,169,3,141,3599  
468 DATA 118,148,141,4,212,173,244,144,2  
05,10,156,16,31,201,65,16,27,238,119,148  
,238,119,148,173,119,3213  
469 DATA 148,201,8,208,11,238,244,144,32  
,212,144,169,0,141,119,148,141,5,212,173  
,244,144,205,10,156,3457  
470 DATA 48,31,201,42,48,27,206,119,148,  
206,119,148,173,119,148,201,254,208,11,2  
06,244,144,32,212,144,3439  
471 DATA 169,6,141,119,148,141,5,212,76,  
159,145,76,65,146,206,117,148,173,117,14  
8,208,245,169,2,141,3282  
472 DATA 117,148,174,116,148,160,0,189,1  
14,150,153,88,122,189,162,150,153,96,122  
,189,210,150,153,56,122,3431  
473 DATA 189,2,151,153,64,122,173,10,210  
,153,192,122,189,50,151,153,72,123,189,9  
8,151,153,80,123,189,3262  
474 DATA 146,151,153,88,123,189,194,151,  
153,96,123,189,242,151,153,40,123,153,10  
4,123,153,112,123,153,120,3506  
475 DATA 123,189,34,152,153,48,123,189,8  
2,152,153,56,123,189,130,152,153,64,123,  
189,178,152,153,128,123,3311  
476 DATA 153,160,123,153,168,123,153,176  
,123,189,226,152,153,136,123,189,18,153,  
153,144,123,189,66,153,153,3652  
477 DATA 152,123,232,200,192,8,208,15,14  
2,116,148,224,48,208,11,169,0,141,116,14  
8,76,65,146,76,177,3141  
478 DATA 145,172,191,122,162,6,189,184,1  
22,157,185,122,202,224,255,208,245,140,1  
84,122,32,129,127,169,63,3857  
479 DATA 160,252,174,112,148,208,6,141,1  
0,122,140,18,122,224,1,208,6,141,11,122,  
140,19,122,224,2,2833  
480 DATA 208,6,141,12,122,140,20,122,169  
,51,160,204,224,3,208,6,141,12,122,140,2  
0,122,224,4,208,2789  
481 DATA 6,141,11,122,140,19,122,224,5,2  
08,6,141,10,122,140,18,122,238,112,148,1  
73,10,210,208,6,2662  
482 DATA 141,114,148,141,112,148,206,115  
,148,173,115,148,201,6,240,71,201,0,208,  
77,173,114,148,208,40,3396  
483 DATA 169,12,141,115,148,173,10,210,2  
01,1,16,5,169,1,141,114,148,169,7,141,25  
,122,169,208,141,2756  
484 DATA 33,122,169,20,141,30,122,169,0,  
141,31,122,76,5,147,169,19,141,25,122,16

9,196,141,33,122,2465  
485 DATA 173,10,210,201,50,16,5,169,0,14  
1,114,148,169,12,141,30,122,169,20,141,3  
1,122,174,22,156,2546  
486 DATA 224,8,240,14,169,1,141,0,210,18  
9,45,148,141,1,210,238,22,156,206,44,148  
,240,3,76,204,3078  
487 DATA 147,169,2,141,44,148,174,28,156  
,224,6,208,39,173,10,210,41,10,105,15,14  
1,0,210,173,10,2584  
488 DATA 210,41,5,105,20,141,2,210,169,1  
70,141,1,210,141,3,210,169,0,141,5,210,1  
41,7,210,76,2738  
489 DATA 246,147,174,23,156,224,16,240,1  
6,169,80,141,2,210,189,53,148,105,32,141  
,3,210,238,23,156,3142  
490 DATA 174,27,156,224,9,240,17,169,50,  
141,4,210,189,103,148,141,5,210,238,27,1  
56,76,204,147,174,3239  
491 DATA 26,156,224,8,240,20,169,4,141,4  
,210,189,95,148,141,5,210,141,200,2,238,  
26,156,76,204,3033  
492 DATA 147,174,24,156,224,16,16,17,169  
,60,141,4,210,189,69,148,141,5,210,238,2  
4,156,76,204,147,2965  
493 DATA 174,25,156,224,10,16,14,169,200  
,141,4,210,189,85,148,141,5,210,238,25,1  
56,173,21,156,240,3130  
494 DATA 17,201,3,240,33,169,164,141,7,2  
10,169,0,141,21,156,76,236,147,169,4,141  
,7,210,169,1,2832  
495 DATA 141,21,156,173,10,210,41,155,10  
5,100,141,6,210,173,40,148,208,42,32,34,  
155,173,159,144,205,2982  
496 DATA 42,148,208,3,32,126,154,173,43,  
148,201,1,208,21,173,159,144,205,173,154  
,208,13,173,160,144,3214  
497 DATA 205,173,154,208,5,169,0,141,27,  
156,76,98,228,0,15,25,50,2,1,2,4,6,8,10,  
6,1769  
498 DATA 0,140,139,138,139,137,138,136,1  
37,133,135,133,132,131,130,129,128,143,1  
43,142,142,141,138,137,136,3277  
499 DATA 135,134,133,132,131,130,129,128  
,170,168,169,166,167,164,165,162,163,160  
,14,12,10,8,6,4,2,2762  
500 DATA 0,170,169,168,167,166,165,163,1  
62,160,0,0,1,12,0,4,0,0,2,169,120,141,24  
4,2,32,2217  
501 DATA 0,129,169,0,133,223,141,40,148,  
141,166,155,173,120,2,201,11,208,60,32,1  
82,153,206,56,150,2999  
502 DATA 206,56,150,32,114,153,173,181,1  
53,201,1,240,17,169,0,141,22,156,32,254,  
149,32,9,150,206,2997  
503 DATA 11,156,206,11,156,32,58,150,173  
,243,144,105,10,205,11,156,48,8,173,243,  
144,233,14,141,9,2840  
504 DATA 156,76,162,149,201,7,208,60,32,  
182,153,238,56,150,238,56,150,32,114,153  
,173,181,153,201,1,3282  
505 DATA 240,17,169,0,141,22,156,32,254,  
149,32,9,150,238,11,156,238,11,156,32,86  
,150,173,243,144,3009



506 DATA 105,34,205,11,156,16,200,173,24  
3,144,105,14,141,9,156,76,162,149,201,14  
208,60,32,182,153,2949  
507 DATA 206,57,150,206,57,150,32,114,15  
3,173,181,153,201,1,240,17,169,0,141,22,  
156,32,254,149,32,3046  
508 DATA 9,150,206,12,156,206,12,156,32,  
86,150,173,244,144,105,6,205,12,156,48,9  
5,173,244,144,233,3157  
509 DATA 8,141,10,156,76,162,149,201,13,  
208,77,32,182,153,238,57,150,238,57,150,  
32,114,153,173,181,3111  
510 DATA 153,201,1,240,34,169,0,141,22,1  
56,32,254,149,32,9,150,32,254,149,169,1,  
129,212,165,212,3066  
511 DATA 141,249,149,165,213,141,250,149  
238,12,156,238,12,156,32,58,150,173,244  
144,105,14,205,12,156,3562  
512 DATA 16,14,173,244,144,105,8,141,10,  
156,76,162,149,32,28,150,32,0,135,173,24  
9,149,133,212,173,2864  
513 DATA 250,149,133,213,162,0,161,212,2  
01,1,208,4,169,0,129,212,169,0,141,249,1  
49,173,252,2,201,3540  
514 DATA 28,240,22,173,166,155,240,3,76,  
69,155,165,223,240,40,169,255,141,43,148

12,12,20,192,64,192,64,252,1,1,860  
527 DATA 1,192,64,192,64,240,4,4,4,192,6  
4,192,64,192,16,16,16,192,64,192,64,192,  
64,64,64,2413  
528 DATA 192,64,192,64,192,16,16,16,192,  
64,192,64,240,4,4,4,224,248,254,191,47,1  
91,255,255,248,3429  
529 DATA 254,255,191,47,191,255,255,254,  
255,255,175,11,175,255,255,254,255,255,1  
75,11,175,255,255,248,254,5220  
530 DATA 255,191,47,191,255,255,224,248,  
254,191,47,191,255,255,2,11,47,190,248,2  
54,255,255,11,47,191,4370  
531 DATA 250,224,250,255,255,47,191,255,  
250,234,255,255,255,47,191,255,250,234,2  
55,255,255,11,47,191,250,5217  
532 DATA 224,250,255,255,2,11,47,190,248  
254,255,255,191,47,11,47,191,254,248,22  
4,255,191,47,191,255,4398  
533 DATA 255,254,248,255,191,191,191,255  
255,255,255,255,191,191,191,255,255,255  
255,255,191,47,191,255,255,5647  
534 DATA 254,248,191,47,11,47,191,254,24  
8,224,254,248,224,248,186,47,11,2,255,25  
4,248,254,255,191,47,4439  
535 DATA 11,255,254,254,254,255,255,191,

**Nur noch wenige Seiten und Sie können sich als (Atari)-„Steinbeißer“ betätigen!**

,173,132,2,208,197,3463  
515 DATA 169,255,141,252,2,206,180,144,1  
73,180,144,205,173,154,208,6,32,187,155,  
76,0,144,32,187,155,3560  
516 DATA 76,189,144,0,98,76,139,148,173,  
11,156,133,212,173,12,156,133,213,96,169  
0,162,0,129,212,3010  
517 DATA 230,212,129,212,230,213,129,212  
198,212,129,212,96,32,254,149,162,0,169  
65,129,212,230,212,169,4197  
518 DATA 66,129,212,230,213,169,68,129,2  
12,198,212,169,67,129,212,96,0,0,32,254,  
149,162,0,169,69,3346  
519 DATA 129,212,230,212,169,70,129,212,  
230,213,169,72,129,212,198,212,169,71,12  
9,212,96,32,254,149,162,4072  
520 DATA 0,169,73,129,212,230,212,169,74  
129,212,230,213,169,76,129,212,198,212,  
169,75,129,212,96,3,3732  
521 DATA 1,3,1,63,64,64,64,3,1,3,1,15,16  
16,16,3,1,3,1,3,4,4,4,3,1,358  
522 DATA 3,1,3,1,1,1,3,1,3,1,3,4,4,4,3,1  
3,1,15,16,16,16,208,64,192,568  
523 DATA 112,12,12,12,5,208,64,192,64,48  
48,48,40,208,64,192,64,192,192,160,  
208,64,192,64,2657  
524 DATA 0,0,0,0,208,64,192,64,192,192,1  
92,160,208,64,192,64,48,48,48,40,7,1,3,1  
3,48,2048  
525 DATA 48,48,80,7,1,3,1,12,12,12,20,7,  
1,3,1,3,3,3,5,7,1,3,1,0,0,282  
526 DATA 0,0,7,1,3,1,3,3,3,5,7,1,3,1,12,

47,255,254,254,254,255,255,191,47,255,25  
4,248,254,255,191,47,11,5056  
536 DATA 254,248,224,248,186,47,11,2,170  
170,128,128,133,133,132,132,170,170,128  
128,133,133,132,132,0,3472  
537 DATA 0,21,21,26,26,24,24,0,0,21,21,2  
6,26,24,24,85,85,106,106,96,96,96,96,85,  
85,1220  
538 DATA 106,106,96,96,96,96,170,170,2,2  
82,82,18,18,170,170,2,2,82,82,18,18,0,0  
84,1768  
539 DATA 84,164,164,36,36,0,0,84,84,164,  
164,36,36,85,85,169,169,9,9,9,9,85,85,16  
9,169,2104  
540 DATA 9,9,9,9,132,132,133,133,128,128  
170,170,132,132,133,133,128,128,170,170  
24,24,26,26,21,2409  
541 DATA 21,0,0,24,24,26,26,21,21,0,0,96  
96,96,96,106,106,85,85,96,96,96,96,106,  
106,1525  
542 DATA 85,85,18,18,82,82,2,2,170,170,1  
8,18,82,82,2,2,170,170,36,36,164,164,84,  
84,0,1826  
543 DATA 0,36,36,164,164,84,84,0,0,9,9,9  
9,169,169,85,85,9,9,9,9,169,169,85,85,1  
656  
544 DATA 64,128,80,88,164,252,1,0,64,128  
80,88,164,252,1,0,64,128,80,88,164,252,  
1,0,0,2331  
545 DATA 0,16,24,36,60,1,0,0,0,0,8,4,12,  
1,0,0,0,16,24,36,60,1,0,2,1,302  
546 DATA 5,53,42,63,0,192,2,1,5,53,42,63  
0,192,2,1,5,53,42,63,0,192,0,0,4,1075



```

547 DATA 52,40,60,0,192,0,0,0,48,32,48,0
,192,0,0,4,52,40,60,0,192,253,0,84,88,14
37
548 DATA 144,240,128,192,253,0,84,88,144
,240,128,192,253,0,84,88,144,240,128,192
,61,0,20,24,16,3083
549 DATA 48,0,0,13,0,4,8,0,0,0,0,61,0,20
,24,16,48,0,0,63,192,53,53,10,15,628
550 DATA 2,3,63,192,53,53,10,15,2,3,63,1
92,53,53,10,15,2,3,60,192,52,52,8,12,0,1
163
551 DATA 0,48,192,48,48,0,0,0,0,60,192,5
2,52,8,12,0,0,173,56,150,133,214,173,57,
150,1818
552 DATA 133,215,162,0,161,214,201,77,24
0,43,201,86,240,39,201,233,240,35,201,82
,240,65,201,225,240,3975
553 DATA 47,201,88,208,5,169,1,141,166,1
55,230,214,161,214,201,102,240,10,201,11
3,240,6,169,0,141,3423
554 DATA 181,153,96,169,1,141,181,153,96
,0,173,11,156,141,56,150,173,12,156,141,
57,150,96,169,0,2812
555 DATA 141,25,156,32,174,154,32,216,15
4,76,169,153,173,10,210,201,10,16,215,17
3,120,2,201,11,208,3032
556 DATA 3,76,53,154,201,7,208,201,230,2
14,230,214,162,0,161,214,208,191,173,56,
150,133,214,169,80,3702
557 DATA 141,93,147,169,0,141,101,147,14
1,23,156,129,214,230,214,129,214,230,214
,169,82,129,214,230,214,3871
558 DATA 169,83,129,214,173,56,150,133,2
14,230,215,169,0,129,214,230,214,129,214
,230,214,169,84,129,214,4105
559 DATA 230,214,169,85,129,214,76,169,1
53,76,175,153,198,214,198,214,162,0,161,
214,208,243,173,56,150,4034
560 DATA 133,214,230,214,169,80,141,93,1
47,169,0,141,101,147,141,23,156,129,214,
198,214,129,214,198,214,3809
561 DATA 169,83,129,214,198,214,169,82,1
29,214,230,215,169,84,129,214,230,214,16
9,85,129,214,230,214,169,4296
562 DATA 0,129,214,230,214,129,214,76,16
9,153,173,173,154,141,159,144,141,160,14
4,141,161,144,169,153,141,3826
563 DATA 6,144,169,154,141,7,144,169,1,1
33,223,96,0,0,0,0,239,245,244,0,239,230,
0,244,233,3061
564 DATA 237,229,0,0,0,0,0,16,162,18,254
,149,144,189,149,144,205,214,154,208,26,
173,215,154,157,3197
565 DATA 149,144,202,224,16,208,11,173,1
80,144,205,42,148,240,3,238,180,144,224,
13,208,219,96,154,144,3709
566 DATA 162,8,254,149,144,189,147,144,2
05,214,154,208,11,173,215,154,157,149,14
4,202,224,5,208,234,173,4029
567 DATA 150,144,205,157,144,208,41,173,
149,144,205,156,144,208,33,169,0,141,26,
156,173,17,156,133,212,3444
568 DATA 173,18,156,133,213,162,0,169,88
,129,212,230,212,129,212,230,213,129,212

```

```

,198,212,129,212,96,206,4073
569 DATA 43,148,208,29,169,50,141,43,148
,162,12,222,149,144,189,149,144,205,41,1
48,208,11,173,42,148,3126
570 DATA 157,149,144,202,224,9,208,234,9
6,169,3,141,21,156,141,40,148,32,142,155
,32,174,154,32,44,3007
571 DATA 155,173,159,144,205,42,148,208,
231,173,173,154,141,159,144,141,160,144,
141,161,144,169,0,141,7,3617
572 DATA 210,173,13,156,105,15,141,13,15
6,238,187,144,173,187,144,205,5,156,208,
6,173,6,156,141,187,3298
573 DATA 144,32,187,155,76,189,144,162,4
0,169,170,141,7,210,173,161,144,105,50,1
41,6,210,160,255,136,3367
574 DATA 208,253,202,208,248,96,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,1215
575 DATA 0,0,174,13,156,189,80,131,141,9
,156,189,81,131,233,56,141,10,156,160,14
,169,6,141,28,2564
576 DATA 156,169,42,133,215,169,20,133,2
14,162,0,173,10,210,16,4,169,87,129,214,
230,214,165,214,201,3449
577 DATA 102,208,237,230,215,165,215,201
,88,208,225,136,208,218,169,0,141,1,210,
141,3,210,141,28,156,3856
578 DATA 96,49,33,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,178
579 DATA 0,0,0,224,2,225,2,0,144
600 REM 597

```

```

1 REM *****
2 REM *
3 REM * BOULDER DASH TEIL 3 *
4 REM * SETZ DIE DATENFILES 1 UND 2 *
5 REM * ZUSAMMEN *
6 REM * (C)1986 VON OLIVER CYRANKA *
7 REM *
8 REM *****
9 REM
10 POKE 106,80:GRAPHICS 0:S=0:FOR I=1536
TO 1617:READ A:S=S+A:POKE I,A:NEXT I
20 IF S<>4869 THEN ? "TIPFERLER IN DEN
DATAS":END
30 REM **** DATENFILE1 LESEN ****
35 ? "BITTE ZU DATENFILE1 VORSPULEN UND"
:? "DIE PLAY-TASTE DRUECKEN.":? "DANN BI
TTE EINE TASTE DRUECKEN."
40 OPEN #1,4,0,"C":GET #1,B1:GET #1,B2:
IF B1<>255 OR B2<>255 THEN ? "FALSCHES D
ATENFILE":END
50 GET #1,B1:GET #1,B2:GET #1,B3:GET #1,
B3:AD=B1+B2*256:TRAP 70:POKE 559,0
60 GET #1,A:POKE AD,A:AD=AD+1:GOTO 60
70 CLOSE #1:POKE 559,34:?:? "DATENFILE
1 IST GELESEN."
80 REM **** DATENFILE2 LESEN ****
90 ? "NUN BITTE ZU DATENFILE2 VORSPULEN"
:? "UND DIE PLAY TASTE DRUECKEN.":? "DAN
N DRUECKEN SIE BITTE EINE TASTE."
100 OPEN #1,4,0,"C":TRAP 120:POKE 559,0
110 GET #1,A:POKE AD,A:AD=AD+1:GOTO 110

```





# ATARI



## Die letzte Seite der „harten“ Eintipparbeit liegt vor Ihnen!

```
120 CLOSE #1:POKE 559,34:?:? "DATENFILE
2 IST AUCH GELESEN."
130 REM *** DAS PROGRAMM SPEICHEN ***
140 ? :? :?
150 ? "PROGRAMM SPEICHERN.":? "BITTE SPU
LEN SIE ZU EINER FREIEN":? "STELLE VOR,
DRUECKEN SIE RECORD+PLAY"
160 ? "UND DANN IRGEND EINE TASTE."
170 X=USR(1536)
180 REM
190 REM **** MASCHINENPROGRAMM ****
200 DATA 104,32,11,6,32,40,6,32,71,6,96,
162,32,169,3,157,66,3,169,37,157
210 DATA 68,3,169,6,157,69,3,169,8,157,7
4,3,32,86,228,96,67,58,155,162,32
220 DATA 169,11,157,66,3,169,0,157,68,3,
169,91,157,69,3,169,31,157,72,3,169
230 DATA 65,157,73,3,32,86,228,96,162,32
,169,12,157,66,3,32,86,228,96
```

```
1 REM *****
2 REM *
3 REM *      BOULDER DASH      *
4 REM *      TEIL 1            *
5 REM * ERZEUGT FILE1 AUF KASSETTE *
6 REM *      (C) 1986          *
7 REM *      VON OLIVER CYRANKA *
8 REM *
9 REM *****
10 ? CHR$(125);"DATENFILE 1 SCHREIBEN.":
? :? :? "1. KASSETTE EINLEGEN":? "2. REC
ORD + PLAY DRUECKEN"
15 ? "3. RETURN DRUECKEN":OPEN #5,8,0,"C
":TRAP 60:POKE 559,0
20 FOR I=300 TO 688:S=0:FOR J=1 TO 25:RE
AD A:PUT #5,A:S=S+A:NEXT J:READ A:IF A=S
THEN NEXT I:GOTO 50
30 ? "DATENFEHLER IN ZEILE ";I
40 POKE 559,34:CLOSE #5:END
50 POKE 559,34:CLOSE #5:?:? "FERTIG. ":
END
60 ER=PEEK(195)
70 IF ER=6 THEN ? "FEHLER IN ZEILE ";I:G
OTO 40
80 ? "FEHLER ";ER;" IN ZEILE ";PEEK(186)
+PEEK(187)*256:GOTO 40
120 REM
130 REM **** DATEN FUER DAS FILE ****
140 REM
```

```
1 REM *****
2 REM *
3 REM *      BOULDER DASH      *
4 REM *      TEIL 2            *
5 REM * ERZEUGT FILE2 AUF KASSETTE *
6 REM *      (C) 1986          *
```

```
7 REM *      VON OLIVER CYRANKA      *
8 REM *
9 REM *****
10 ? CHR$(125);"DATENFILE 2 SCHREIBEN.":
? :? :? "BITTE KASSETTE EINLEGEN + RETUR
N"
15 TRAP 60:OPEN #5,8,0,"C":POKE 559,0
20 FOR I=301 TO 579:S=0:FOR J=1 TO 25:RE
AD A:PUT #5,A:S=S+A:NEXT J:READ A:IF S=A
THEN NEXT I
30 ? "DATENFEHLER IN ZEILE ";I
40 POKE 559,34:CLOSE #5:END
50 CLOSE #5:?:? :? "FERTIG.":POKE 559,3
4:END
60 ER=PEEK(195)
70 IF ER=6 AND I=579 THEN 50
80 IF ER=8 THEN ? "FEHLER IN ZEILE ";I:G
OTO 40
90 ? "FEHLER ";ER;" IN ZEILE ";PEEK(186)
+PEEK(187)*256:GOTO 40
120 REM
130 REM **** DATEN FUER DAS FILE ****
140 REM
```

```
1 REM *****
2 REM *
3 REM *      BOULDER DASH      *
4 REM * LADERROUTINE FUER KASSETTE *
5 REM *      (C) 1986          *
6 REM *      VON OLIVER CYRANKA *
7 REM *
8 REM *****
9 REM
10 POKE 106,80:GRAPHICS 17:DL=PEEK(560)+
PEEK(561)*256:POKE 559,0
20 POKE DL+9,7:POKE DL+15,2:POKE 710,0:P
OKE 708,118:POKE 709,54
30 POSITION 4,4:?:? #6;"boulder dash":POSI
TION 0,6:?:? #6;"LADEZEIT CA. 12 MIN."
40 POSITION 2,10:?:? #6;"KASSETTE EINLEGEN
UND PLAY DRUECKEN."
50 S=0:FOR I=1536 TO 1619:READ A:POKE I,
A:S=S+A:NEXT I
60 IF S<>6993 THEN GRAPHICS 0:?:? "TIPFEL
ER IN DEN DATAS":END
70 POKE 559,34:X=USR(1536)
80 REM
90 REM **** LADERROUTINE FUER CID ****
100 DATA 104,32,13,6,32,42,6,32,73,6,76,
0,144,162,32,169,3,157,66,3,169
110 DATA 39,157,68,3,169,6,157,69,3,169,
4,157,74,3,32,86,228,96,67,58,155
120 DATA 162,32,169,7,157,66,3,169,0,157
,68,3,169,91,157,69,3,169,31,157,72
130 DATA 3,169,65,157,73,3,32,86,228,96,
162,32,169,12,157,66,3,32,86,228,96
```

*Sie werden feststellen, daß „Boulder Dash“ ein echtes Topprogramm ist!*

# Zone - M

**Für den ZX-Spectrum**  
**mit 16 oder 48 K**

## Steuerung des Spieles:

Sie können das Spiel entweder mit dem Kempston-Joystick oder mit den Tasten (P, I, Q und Z) steuern. Schießen können Sie mit der Taste (N) oder durch das Drücken des Feuerknopfes (Joysticks).

Weiterhin:

- (1) hält das Spiel an,
- (2) setzt das Spiel nach (1) wieder fort und
- (3) bricht das Spiel ab, und kehrt zum Titelbild zurück.

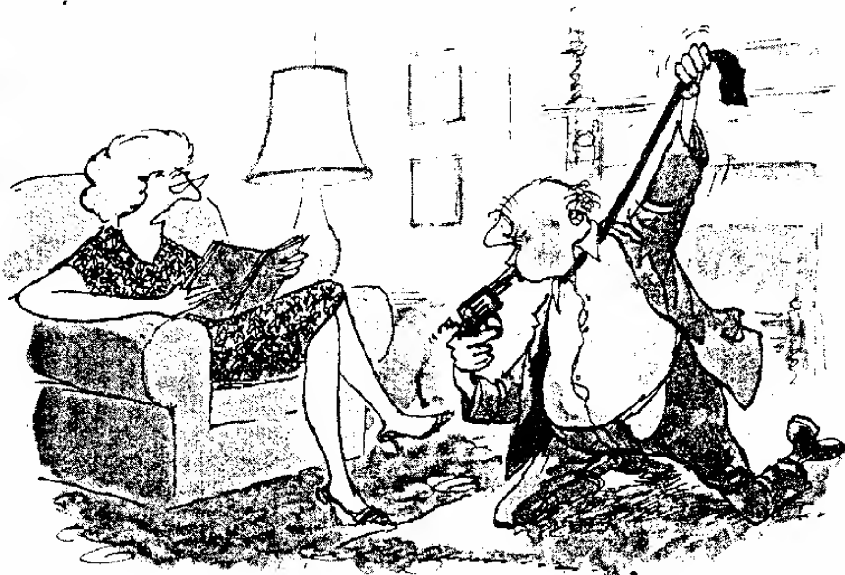
## Eingabe und Speicherung des Programms:

1. Geben Sie das kurze Basicprogramm „Zone-M“ ein und speichern es durch: SAVE „Zone-M“ LINE 1 auf Kassette ab.
2. Geben Sie nun das Programm „Eingabe“ ein und starten es.
3. Geben Sie die Zahlenblöcke der Reihe nach ein, und vergleichen Sie nach jedem eingegebenen Block die Prüfsumme.
4. Bei Ungleichheit brechen Sie das Programm sofort mit STOP ab, und starten es erneut. Geben Sie diesmal als Anfangsadresse die Anfangsadresse des fehlerhaften Blocks ein, und tippen Sie den letzten Zahlenblock nochmal ein.
5. Nach Beendigung der Eingabe speichern Sie das Maschinenprogramm hinter dem Basic-Lader auf Kassette ab. SAVE „ZM-mc“ CODE 24576,3660.

Ein rasend schnelles Action-Spiel mit 13 (!) verschiedenen Schwierigkeitsstufen und tollen Effekten!

Die Spielaufgabe von „ZONE - M“ ist schnell erläutert: Überstehen Sie die 13

verschiedenen Spielstufen, indem Sie Ihre immer zahlreicher werdenden Gegner abschießen. Sollte Ihnen dies nicht gelingen, so wird Ihnen selbst durch Ihre Widersacher der „Gar aus gemacht“.



*Verzweifeln Sie nicht, wenn Sie anfangs der Spielaufgabe nicht gewachsen sind. Nach einiger Übung wird auch für Sie das Erreichen des 13. Levels ein realistisches Ziel sein!*

## ZONE-M

## Das Listing

```

1 REM
2 REM
3 REM   Zone-M-Eingabeprogr.
4 REM   1986 von:
5 REM
6 REM   Zuheir Urwani
7 REM
8 REM
100
110 CLEAR 24575
120 INPUT "Anfangsadr.:";a
130 FOR i=a TO 28235 STEP 60
140 LET S=a-a
150 FOR j=i TO i+59 STEP 6
160 PRINT j;" :";
170 FOR n=j TO j+5
180 INPUT (n;" : ");u
190 POKE n,u: LET S=S+u
200 PRINT " ( TO 4-LEN STRS
u);u;
210 NEXT n: PRINT **: NEXT j
    
```

```

220 PRINT "'Summe:';S': NEXT i
230 SAVE "ZM-mc"CODE 24576,3660
    
```

```

1 REM
2 REM   ZONE-M
3 REM
4 REM   Hauptprogramm
5 REM
6 REM   1986 Zuheir Urwani
7 REM
10 CLEAR 24575
20 LOAD "ZM-mc"CODE 24576,3660
30 RANDOMIZE USR 25064
40 STOP
50 SAVE "ZONE-M" LINE 10
60 SAVE "ZM-mc"CODE 24576,3660
    
```

```

24576 : 0 0 0 0 0 0
24582 : 0 0 14 14 14 28
24588 : 24 16 192 192 0 24
24594 : 24 0 0 24 24 0
24600 : 0 0 0 0 0 24
24606 : 24 0 0 0 0 0
24612 : 24 24 8 48 124 130
24618 : 186 162 162 186 130 124
24624 : 8 8 8 255 8 8
24630 : 8 8 0 0 0 255
    
```

Summe : 2512

```

24636 : 0 0 0 0 0 24
24642 : 60 60 60 60 24 0
24648 : 0 0 60 126 126 60
24654 : 0 0 24 24 189 189
24660 : 189 255 153 189 16 76
24666 : 184 62 124 90 160 20
24672 : 0 24 24 60 90 165
    
```



24678 : 189 99 6 9 23 122  
24684 : 122 23 9 6 99 189  
24690 : 165 90 60 24 24 0

Summe : 4226

24696 : 96 144 232 94 94 232  
24702 : 144 96 0 124 198 206  
24708 : 214 230 198 124 0 56  
24714 : 24 24 24 24 24 60  
24720 : 0 124 142 14 28 32  
24726 : 126 204 0 124 142 14  
24732 : 28 14 142 124 0 28  
24738 : 44 76 140 254 12 30  
24744 : 0 126 64 64 124 6  
24750 : 70 60 0 124 194 192

Summe : 5528

24756 : 252 206 198 124 0 126  
24762 : 252 132 8 24 48 48  
24768 : 0 124 198 198 124 198  
24774 : 198 124 0 124 198 230  
24780 : 126 6 134 124 102 255  
24786 : 255 6 96 255 255 102  
24792 : 24 126 242 15 240 207  
24798 : 126 24 0 0 0 24  
24804 : 24 0 0 0 0 0  
24810 : 24 60 60 24 0 0

Summe : 6070

24816 : 0 24 60 126 126 60  
24822 : 24 0 0 24 60 102  
24828 : 102 60 24 0 60 126  
24834 : 231 195 195 231 126 60  
24840 : 0 24 28 44 38 126  
24846 : 70 70 0 252 102 102  
24852 : 124 102 102 252 0 62  
24858 : 98 96 96 96 98 60  
24864 : 0 248 108 102 102 102  
24870 : 108 248 0 254 98 100

Summe : 5528

24876 : 124 100 98 254 0 254  
24882 : 98 100 124 100 96 240  
24888 : 0 126 226 224 224 238  
24894 : 230 124 0 198 198 198  
24900 : 254 198 198 198 0 120  
24906 : 48 48 48 48 48 120  
24912 : 0 30 12 12 12 108  
24918 : 108 56 0 230 108 104  
24924 : 124 110 102 230 0 240  
24930 : 96 96 96 96 100 252

Summe : 7224

24936 : 0 198 238 254 182 166  
24942 : 134 134 0 226 114 122  
24948 : 94 78 70 230 0 124  
24954 : 198 198 198 198 198 124  
24960 : 0 252 102 102 124 96  
24966 : 96 240 0 124 198 198  
24972 : 198 214 204 118 0 252  
24978 : 102 102 120 108 108 230  
24984 : 0 126 194 112 60 14  
24990 : 134 252 0 252 180 48

Summe : 8138

24996 : 48 48 48 120 0 238  
25002 : 196 196 196 196 204 120  
25008 : 0 206 196 228 100 116  
25014 : 60 24 0 218 218 218  
25020 : 236 236 108 108 0 254  
25026 : 142 28 56 112 226 254  
25032 : 0 198 228 120 56 24  
25038 : 24 24 0 3 0 15  
25044 : 0 63 0 255 0 254  
25050 : 0 248 0 224 0 128

Summe : 6818

25056 : 0 255 0 255 0 255  
25062 : 0 255 175 33 250 127  
25068 : 17 251 127 119 1 5  
25074 : 0 237 176 33 0 95  
25080 : 34 54 92 50 141 92  
25086 : 50 72 92 211 254 62  
25092 : 2 205 1 22 205 107  
25098 : 13 219 31 254 50 48  
25104 : 5 62 1 50 250 127  
25110 : 205 239 109 33 242 127

Summe : 6477

25116 : 17 243 127 54 0 1  
25122 : 7 0 237 176 33 8  
25128 : 5 34 240 127 205 112  
25134 : 103 58 243 127 205 105  
25140 : 107 62 2 205 1 22  
25146 : 14 1 62 67 50 143  
25152 : 92 6 30 62 22 215  
25158 : 121 215 62 1 215 62  
25164 : 39 215 16 251 197 205  
25170 : 86 98 24 14 1 5

Summe : 5457

25176 : 3 10 230 16 211 254  
25182 : 13 32 248 16 246 201  
25188 : 17 5 0 33 0 1  
25194 : 205 181 3 193 12 62  
25200 : 21 185 32 205 14 1  
25206 : 6 1 62 22 215 120  
25212 : 215 121 215 62 38 215  
25218 : 4 62 21 184 32 240  
25224 : 197 205 86 98 0 0  
25230 : 0 0 17 8 0 33

Summe : 5129

25236 : 55 0 205 181 3 193  
25242 : 12 62 31 185 32 214  
25248 : 22 11 62 68 50 143  
25254 : 92 62 42 205 242 98  
25260 : 213 122 135 135 135 135  
25266 : 111 96 17 20 0 205  
25272 : 181 3 205 86 98 209  
25278 : 62 67 50 143 92 62  
25284 : 38 205 242 98 21 32  
25290 : 215 62 68 50 143 92

Summe : 6118

25296 : 62 22 215 62 11 215  
25302 : 62 16 215 62 42 215  
25308 : 62 1 205 1 22 62  
25314 : 69 50 143 92 17 43

25320 : 99 1 29 0 205 60  
25326 : 32 195 72 99 245 14  
25332 : 255 30 255 121 179 254  
25338 : 0 40 34 33 16 11  
25344 : 66 125 129 111 124 131  
25350 : 103 16 248 124 61 254

Summe : 5737

25356 : 20 48 16 125 61 254  
25362 : 30 48 10 62 22 215  
25368 : 124 215 125 215 241 245  
25374 : 215 28 62 2 187 32  
25380 : 210 12 185 32 204 241  
25386 : 201 22 1 0 83 67  
25392 : 34 22 1 11 88 79  
25398 : 78 69 34 22 1 23  
25404 : 72 73 34 22 1 18  
25410 : 16 4 42 16 5 34

Summe : 4659

25416 : 62 1 205 1 22 62  
25422 : 71 50 143 92 62 22  
25428 : 215 62 1 215 62 20  
25434 : 215 58 241 127 198 48  
25440 : 215 62 22 215 62 1  
25446 : 215 62 16 215 58 243  
25452 : 127 198 65 215 33 251  
25458 : 127 22 26 205 130 99  
25464 : 33 245 127 22 3 205  
25470 : 130 99 24 21 62 22

Summe : 6132

25476 : 215 62 1 215 122 215  
25482 : 6 5 126 35 198 48  
25488 : 215 16 249 62 48 215  
25494 : 201 62 247 219 254 230  
25500 : 31 254 1 200 71 230  
25506 : 1 32 8 62 247 219  
25512 : 254 230 2 32 248 120  
25518 : 230 4 202 22 98 33  
25524 : 0 2 43 124 181 254  
25530 : 0 32 249 0 58 244

Summe : 7284

25536 : 127 60 230 7 50 244  
25542 : 127 205 135 100 254 1  
25548 : 40 31 205 104 103 254  
25554 : 1 40 24 205 228 104  
25560 : 205 218 103 254 1 40  
25566 : 14 254 2 40 64 205  
25572 : 62 106 254 4 40 3  
25578 : 195 72 99 58 241 127  
25584 : 254 0 40 13 61 50  
25590 : 241 127 17 100 100 205

Summe : 6748

25596 : 133 106 195 44 98 17  
25602 : 80 100 205 133 106 1  
25608 : 251 127 33 245 127 229  
25614 : 197 10 190 32 4 3  
25620 : 35 24 248 209 225 48  
25626 : 5 1 5 0 237 176  
25632 : 195 22 98 33 243 127  
25638 : 126 254 12 40 31 52



25644 : 43 126 203 127 32 4  
25650 : 246 128 24 1 7 119

Summe : 6172

25656 : 43 126 254 9 40 1  
25662 : 52 43 52 17 90 100  
25668 : 205 231 106 195 44 98  
25674 : 205 117 106 195 66 110  
25680 : 71 65 77 69 32 32  
25686 : 79 86 69 82 88 79  
25692 : 78 69 32 32 79 86  
25698 : 69 82 68 69 83 84  
25704 : 82 79 89 69 68 33  
25710 : 229 175 203 60 31 203

Summe : 5486

25716 : 60 31 203 60 31 181  
25722 : 111 124 198 88 103 126  
25728 : 214 64 225 201 0 0  
25734 : 0 58 244 127 230 3  
25740 : 254 0 200 62 2 205  
25746 : 1 22 14 0 62 223  
25752 : 219 254 203 71 32 4  
25758 : 14 1 24 6 203 87  
25764 : 32 2 14 2 62 254  
25770 : 219 254 230 2 32 4

Summe : 5952

25776 : 203 209 24 10 62 251  
25782 : 219 254 230 1 32 2  
25788 : 203 217 62 127 219 254  
25794 : 230 8 32 2 203 225  
25800 : 175 185 32 10 58 250  
25806 : 127 254 0 40 3 219  
25812 : 31 79 42 238 127 229  
25818 : 209 203 65 40 1 44  
25824 : 203 73 40 1 45 203  
25830 : 81 40 1 36 203 89

Summe : 6955

25836 : 40 1 37 121 230 15  
25842 : 0 40 65 124 214 4  
25848 : 254 17 56 1 98 125  
25854 : 61 254 30 56 1 107  
25860 : 195 123 101 0 0 254  
25866 : 4 40 41 254 3 40  
25872 : 2 203 249 62 67 50  
25878 : 143 92 62 22 215 122  
25884 : 215 123 215 62 38 215  
25890 : 62 68 50 143 92 62

Summe : 5642

25896 : 22 215 124 215 125 215  
25902 : 62 42 215 229 34 238  
25908 : 127 209 203 121 40 3  
25914 : 62 1 201 203 97 32  
25920 : 2 175 201 58 19 126  
25926 : 254 0 192 21 33 19  
25932 : 126 54 255 35 114 35  
25938 : 115 35 54 1 62 71  
25944 : 50 143 92 62 22 215  
25950 : 122 215 123 215 205 138

Summe : 6694

25956 : 101 6 60 17 1 0  
25962 : 33 0 0 213 229 197  
25968 : 205 181 3 193 225 209  
25974 : 35 16 244 175 201 122  
25980 : 188 32 5 123 189 202  
25986 : 54 101 205 110 100 195  
25992 : 7 101 98 107 205 110  
25998 : 100 254 3 192 62 40  
26004 : 215 201 175 50 176 92  
26010 : 221 33 19 126 221 126

Summe : 7104

26016 : 0 254 0 40 50 254  
26022 : 255 40 99 79 221 126  
26028 : 3 61 221 119 3 32  
26034 : 36 12 62 6 185 221  
26040 : 113 0 32 52 221 54  
26046 : 0 255 221 54 3 3  
26052 : 62 22 215 221 126 1  
26058 : 215 221 126 2 215 62  
26064 : 71 50 143 92 62 40  
26070 : 215 1 4 0 221 9

Summe : 5783

26076 : 62 1 50 176 92 221  
26082 : 229 193 62 127 184 32  
26088 : 181 62 163 185 32 249  
26094 : 175 201 221 54 3 64  
26100 : 62 22 215 221 126 1  
26106 : 215 221 126 2 215 62  
26112 : 70 50 143 92 221 126  
26118 : 0 198 59 215 24 203  
26124 : 221 102 1 221 110 2  
26130 : 205 110 100 254 7 40

Summe : 7281

26136 : 4 254 3 32 100 62  
26142 : 67 50 143 92 62 22  
26148 : 215 124 215 125 215 62  
26154 : 38 215 221 126 3 254  
26160 : 3 40 17 254 2 40  
26166 : 10 254 1 40 3 44  
26172 : 24 7 37 24 4 45  
26178 : 24 1 36 95 124 61  
26184 : 254 20 48 6 125 61  
26190 : 254 30 56 7 221 54

Summe : 5030

26196 : 0 0 195 215 101 205  
26202 : 110 100 254 3 40 4  
26208 : 254 7 32 29 221 116  
26214 : 1 221 117 2 62 71  
26220 : 50 143 92 62 22 215  
26226 : 124 215 125 215 62 40  
26232 : 203 67 32 1 60 215  
26238 : 195 215 101 58 176 92  
26244 : 254 0 40 15 58 238  
26250 : 127 189 32 196 58 239

Summe : 6586

26256 : 127 188 32 248 62 1  
26262 : 201 229 221 229 17 10  
26268 : 0 33 0 2 205 181

26274 : 3 221 225 225 205 110  
26280 : 100 254 6 32 75 17  
26286 : 39 126 26 61 254 5  
26292 : 48 12 19 26 188 32  
26298 : 8 19 26 189 32 4  
26304 : 24 6 19 19 19 19  
26310 : 24 232 19 62 64 18

Summe : 5098

26316 : 27 27 27 62 22 215  
26322 : 124 215 125 215 26 61  
26328 : 18 40 11 62 70 50  
26334 : 143 92 26 198 59 215  
26340 : 24 8 62 67 50 143  
26346 : 92 62 38 215 22 1  
26352 : 30 1 205 80 103 195  
26358 : 82 102 254 4 32 53  
26364 : 50 225 127 30 5 22  
26370 : 2 205 80 103 229 193

Summe : 5326

26376 : 33 39 126 126 254 0  
26382 : 40 6 35 35 35 35  
26388 : 24 245 54 2 35 112  
26394 : 35 113 35 54 64 62  
26400 : 22 215 120 215 121 215  
26406 : 62 70 50 143 92 62  
26412 : 61 215 195 82 102 17  
26418 : 163 127 26 188 32 13  
26424 : 19 26 189 32 9 19  
26430 : 26 254 0 32 7 24

Summe : 4844

26436 : 2 19 19 19 24 234  
26442 : 175 18 30 2 24 177  
26448 : 229 62 250 146 111 38  
26454 : 127 126 131 254 10 48  
26460 : 3 119 225 201 214 10  
26466 : 119 30 1 43 24 239  
26472 : 62 2 205 1 22 195  
26478 : 150 101 33 179 103 17  
26484 : 223 127 1 17 0 237  
26490 : 176 33 227 127 1 2

Summe : 5744

26496 : 4 205 196 103 230 15  
26502 : 60 119 125 129 111 12  
26508 : 16 243 33 19 126 17  
26514 : 20 126 112 1 203 1  
26520 : 237 176 58 240 127 71  
26526 : 135 128 198 162 111 38  
26532 : 127 14 1 54 4 43  
26538 : 113 43 54 1 43 12  
26544 : 16 245 201 1 1 200  
26550 : 21 0 0 0 0 3

Summe : 5104

26556 : 0 31 1 0 0 2  
26562 : 16 11 197 237 95 79  
26568 : 58 120 92 230 31 71  
26574 : 10 79 58 120 92 169  
26580 : 79 237 95 169 193 201  
26586 : 58 244 127 230 1 200  
26592 : 175 50 176 92 221 33  
26598 : 163 127 221 94 2 175





26604 : 187 32 20 221 35 221  
26610 : 35 221 35 221 229 193

Summe : 6812

26616 : 121 254 223 32 235 58  
26622 : 176 92 238 2 201 62  
26628 : 2 50 176 92 221 102  
26634 : 0 221 110 1 62 22  
26640 : 215 124 215 125 215 62  
26646 : 67 50 143 92 62 38  
26652 : 215 123 254 3 40 10  
26658 : 254 2 40 3 44 24  
26664 : 4 45 24 1 36 125  
26670 : 61 254 30 48 80 124

Summe : 6005

26676 : 254 21 32 15 30 4  
26682 : 33 1 1 205 110 100  
26688 : 254 3 40 10 44 24  
26694 : 246 205 110 100 254 3  
26700 : 32 38 123 254 3 32  
26706 : 6 221 126 2 238 6  
26712 : 95 221 116 0 221 117  
26718 : 1 221 115 2 62 22  
26724 : 215 124 215 125 215 62  
26730 : 69 50 143 92 62 58

Summe : 5803

26736 : 215 195 158 104 58 238  
26742 : 127 189 32 9 58 239  
26748 : 127 188 32 3 62 1  
26754 : 201 221 102 0 221 110  
26760 : 1 221 86 2 123 186  
26766 : 32 4 30 3 24 137  
26772 : 254 3 32 202 122 238  
26778 : 6 95 24 244 124 254  
26784 : 21 40 62 58 243 127  
26790 : 203 63 71 205 196 103

Summe : 6729

26796 : 184 48 50 36 17 39  
26802 : 126 26 254 0 40 6  
26808 : 19 19 19 19 24 245  
26814 : 205 110 100 229 193 213  
26820 : 225 54 255 35 112 35  
26826 : 113 35 54 3 254 3  
26832 : 32 15 62 22 215 120  
26838 : 215 121 215 62 71 50  
26844 : 143 92 62 40 215 195  
26850 : 239 103 58 244 127 230

Summe : 6347

26856 : 3 254 0 192 58 242  
26862 : 127 230 15 87 1 19  
26868 : 126 221 33 223 127 3  
26874 : 3 3 3 203 58 208  
26880 : 221 35 221 35 221 35  
26886 : 221 102 0 221 110 1  
26892 : 221 94 2 62 22 215  
26898 : 124 215 125 215 175 50  
26904 : 143 92 62 32 215 123  
26910 : 254 3 40 17 254 2

Summe : 6619

Summe : 6619

26916 : 40 10 254 1 40 3  
26922 : 44 24 23 37 24 4  
26928 : 45 24 17 36 124 61  
26934 : 254 20 56 24 40 4  
26940 : 38 20 24 18 38 1  
26946 : 24 14 125 61 254 30  
26952 : 56 8 40 4 46 30  
26958 : 24 2 46 1 221 116  
26964 : 0 221 117 1 62 22  
26970 : 215 124 215 125 215 62

Summe : 3829

26976 : 71 50 143 92 123 198  
26982 : 44 215 10 254 0 194  
26988 : 249 104 58 242 127 203  
26994 : 111 40 14 58 238 127  
27000 : 189 40 17 58 239 127  
27006 : 188 32 234 24 9 205  
27012 : 196 103 230 15 254 0  
27018 : 32 223 123 203 71 40  
27024 : 6 214 2 133 111 24  
27030 : 3 61 132 103 205 110

Summe : 6921

27036 : 100 254 3 32 18 62  
27042 : 22 215 124 215 125 215  
27048 : 62 71 50 143 92 123  
27054 : 230 1 198 40 215 62  
27060 : 255 2 3 124 2 3  
27066 : 125 2 3 123 60 230  
27072 : 3 2 195 252 104 58  
27078 : 244 127 230 1 200 58  
27084 : 225 127 95 254 4 56  
27090 : 16 28 60 40 5 50

Summe : 6038

27096 : 225 127 175 201 1 1  
27102 : 1 30 3 24 67 33  
27108 : 223 127 70 35 78 62  
27114 : 22 215 120 215 121 215  
27120 : 62 67 50 143 92 62  
27126 : 38 215 22 0 197 225  
27132 : 123 170 203 71 40 3  
27138 : 44 24 1 45 203 79  
27144 : 40 3 36 24 1 37  
27150 : 205 110 100 254 4 200

Summe : 5584

27156 : 254 3 40 9 122 60  
27162 : 230 3 87 32 219 24  
27168 : 2 229 193 123 170 95  
27174 : 33 223 127 112 35 113  
27180 : 35 115 62 22 215 120  
27186 : 215 121 215 62 68 50  
27192 : 143 92 62 59 215 201  
27198 : 58 242 127 230 16 200  
27204 : 195 197 105 229 197 213  
27210 : 33 0 88 17 1 88

Summe : 6846

27216 : 54 0 1 191 2 237  
27222 : 176 209 193 225 201 229  
27228 : 33 191 90 126 254 0  
27234 : 40 9 254 65 32 3

27240 : 175 24 1 61 119 43  
27246 : 124 254 87 32 236 225  
27252 : 201 17 123 106 195 231  
27258 : 106 32 83 85 80 69  
27264 : 82 32 32 33 32 205  
27270 : 71 106 58 238 127 111

Summe : 6651

27276 : 58 239 127 103 14 1  
27282 : 62 71 50 143 92 213  
27288 : 30 255 22 255 229 122  
27294 : 179 254 0 40 28 65  
27300 : 125 130 111 124 131 103  
27306 : 16 248 254 22 48 15  
27312 : 125 254 32 48 10 62  
27318 : 22 215 124 215 125 215  
27324 : 62 43 215 225 20 62  
27330 : 2 186 32 214 28 187

Summe : 6702

27336 : 32 208 205 91 106 197  
27342 : 229 17 30 0 121 135  
27348 : 135 111 38 1 205 181  
27354 : 3 205 86 98 225 193  
27360 : 209 12 121 254 31 32  
27366 : 176 205 71 106 1 3  
27372 : 79 33 21 17 205 214  
27378 : 108 124 215 125 215 197  
27384 : 213 229 1 10 0 205  
27390 : 60 32 225 209 193 124

Summe : 7127

27396 : 214 5 254 13 56 4  
27402 : 121 238 1 79 125 61  
27408 : 254 21 56 4 121 238  
27414 : 2 79 203 65 40 3  
27420 : 36 24 1 37 203 73  
27426 : 40 3 44 24 1 45  
27432 : 197 213 229 205 86 98  
27438 : 33 0 1 17 10 0  
27444 : 205 181 3 225 209 193  
27450 : 205 91 106 16 177 6

Summe : 5494

27456 : 20 197 213 229 62 22  
27462 : 215 124 215 125 215 1  
27468 : 10 0 205 60 32 205  
27474 : 86 98 33 0 1 17  
27480 : 10 0 205 181 3 205  
27486 : 91 106 225 209 193 16  
27492 : 220 205 71 106 201 205  
27498 : 223 108 205 43 108 33  
27504 : 0 88 221 33 48 117  
27510 : 17 92 74 6 32 221

Summe : 6506

27516 : 126 0 203 79 32 8  
27522 : 254 0 32 3 114 24  
27528 : 1 115 221 35 35 16  
27534 : 236 62 91 188 40 40  
27540 : 122 230 71 79 135 135  
27546 : 135 71 121 60 254 71  
27552 : 32 2 62 65 176 87  
27558 : 123 230 56 79 191 31



**Der Spectrum-Computer hat immer noch einiges zu bieten. Dies beweist unser Top-Programm „ZONE-M“, daß aus der (Software)-Feder“ von Zuheir Urwani stammt.**

27564 : 31 31 71 121 214 8  
27570 : 32 2 62 48 176 246

Summe : 5615

27576 : 64 95 24 189 62 22  
27582 : 215 62 11 215 62 13  
27588 : 215 62 71 50 143 92  
27594 : 62 88 215 62 79 215  
27600 : 62 78 215 62 69 215  
27606 : 62 32 215 58 243 127  
27612 : 198 65 215 175 50 176  
27618 : 92 50 177 92 6 60  
27624 : 197 33 48 117 34 48  
27630 : 120 33 0 64 34 50

Summe : 5960

27636 : 120 38 88 34 52 120  
27642 : 205 52 108 33 48 118  
27648 : 34 48 120 33 0 72  
27654 : 34 50 120 38 89 34  
27660 : 52 120 205 52 108 33  
27666 : 48 119 34 48 120 33  
27672 : 0 80 34 50 120 38  
27678 : 90 34 52 120 205 52  
27684 : 108 205 129 108 193 16  
27690 : 189 205 107 13 62 2

Summe : 4872

27696 : 205 1 22 201 243 253  
27702 : 229 253 42 48 120 42  
27708 : 50 120 58 176 92 79  
27714 : 132 229 103 229 221 225  
27720 : 42 52 120 235 175 145  
27726 : 198 7 225 132 103 6  
27732 : 0 253 126 0 254 0  
27738 : 32 6 126 238 255 119  
27744 : 24 19 254 1 32 10  
27750 : 221 126 0 238 255 221

Summe : 7623

27756 : 119 0 24 5 26 60  
27762 : 230 71 18 221 35 253  
27768 : 35 19 35 16 216 253  
27774 : 225 251 201 121 60 230  
27780 : 7 33 176 92 119 194  
27786 : 22 109 35 126 238 1  
27792 : 119 33 48 117 17 0  
27798 : 88 1 3 0 58 177  
27804 : 92 254 0 40 17 126

27810 : 203 79 32 31 26 198

Summe : 5635

27816 : 16 254 126 56 2 214  
27822 : 48 18 24 19 126 203  
27828 : 79 32 14 26 198 2  
27834 : 254 119 40 4 254 80  
27840 : 32 2 214 6 18 35  
27846 : 19 16 209 13 32 206  
27852 : 201 1 24 0 16 254  
27858 : 13 32 251 201 62 71  
27864 : 50 143 92 62 22 215  
27870 : 201 33 48 117 17 49

Summe : 5185

27876 : 117 1 255 2 54 1  
27882 : 237 176 33 114 117 54  
27888 : 0 43 54 0 17 117  
27894 : 117 1 124 2 237 176  
27900 : 33 124 118 6 3 197  
27906 : 229 229 209 19 1 7  
27912 : 0 54 2 237 176 225  
27918 : 1 32 0 9 193 16  
27924 : 236 201 203 71 202 205  
27930 : 108 229 33 150 0 17

Summe : 5824

27936 : 10 0 205 181 3 225  
27942 : 201 0 127 159 159 159  
27948 : 128 127 254 0 6 102  
27954 : 102 102 1 134 24 1  
27960 : 225 153 153 249 230 24  
27966 : 96 24 6 102 102 0  
27972 : 24 97 128 127 159 153  
27978 : 159 128 97 134 22 9  
27984 : 9 16 4 65 78 32  
27990 : 65 67 84 73 79 78

Summe : 5661

27996 : 32 71 65 77 69 22  
28002 : 11 11 16 3 87 82  
28008 : 73 84 84 69 78 32  
28014 : 66 89 22 13 9 16  
28020 : 7 88 85 72 69 73  
28026 : 82 32 32 85 82 87  
28032 : 65 78 73 22 17 0  
28038 : 16 5 67 79 78 84  
28044 : 82 79 76 32 75 69  
28050 : 89 83 34 32 32 16

Summe : 3258

28056 : 6 80 36 73 36 81  
28062 : 36 88 32 65 78 68  
28068 : 32 78 35 22 19 0  
28074 : 16 5 79 82 32 85  
28080 : 83 69 32 84 72 69  
28086 : 32 75 69 77 80 83  
28092 : 84 79 78 32 74 79  
28098 : 89 83 84 73 67 75  
28104 : 35 22 21 0 16 4  
28110 : 80 82 69 83 83 32

Summe : 3413

28116 : 69 78 84 69 82 32  
28122 : 84 79 32 83 84 65  
28128 : 82 84 32 33 22 21  
28134 : 26 16 7 37 32 49  
28140 : 57 56 54 175 50 145  
28146 : 92 62 64 50 141 92  
28152 : 62 2 205 1 22 205  
28158 : 43 108 62 22 215 62  
28164 : 2 215 175 215 62 70  
28170 : 50 143 92 33 39 109

Summe : 4534

28176 : 14 39 94 6 4 175  
28182 : 203 35 23 203 35 23  
28188 : 254 0 32 5 62 32  
28194 : 215 24 3 198 89 215  
28200 : 16 235 35 13 32 228  
28206 : 17 78 109 1 161 0  
28212 : 205 60 32 62 191 219  
28218 : 254 230 1 32 248 195  
28224 : 43 108 17 80 100 205  
28230 : 231 106 195 7 100 0

Summe : 5829

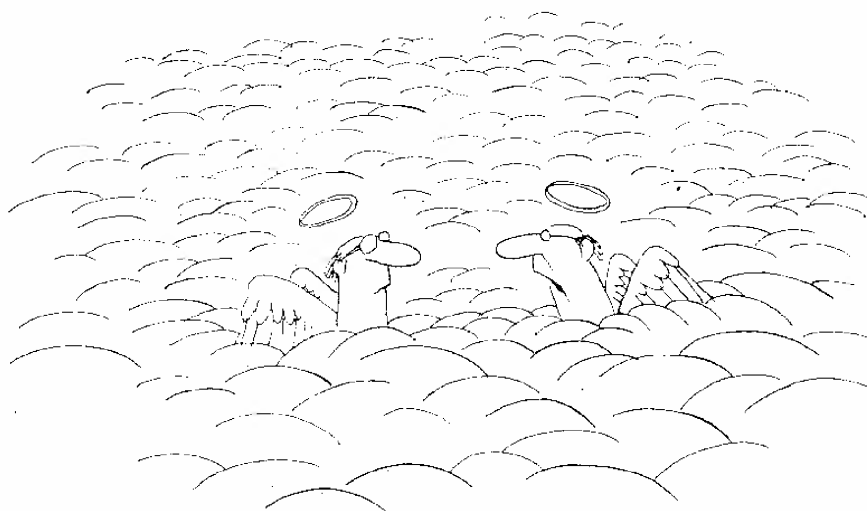
28236 : 0 0 0 0 0 0  
28242 : 0 0 0 0 0 0  
28248 : 0 0 0 0 0 0  
28254 : 0 0 0 0 0 0  
28260 : 0 0 0 0 0 0  
28266 : 0 0 0 0 0 0  
28272 : 0 0 0 0 0 0  
28278 : 0 0 0 0 0 0  
28284 : 0 0 0 0 0 0  
28290 : 0 0 0 0 0 0

Summe : 0

*Ein ZX-Spectrum-Beitrag, der mit Sicherheit Ihre vollste Konzentration und Ihre schnellsten Reaktionen erfordert.*

*Springen Sie durch die Lüfte und sammeln Sie dabei Luftballons ein. Zeigen Sie, daß Sie die Befähigung haben ein guter Akrobat zu werden.*

# Jumpman für den APPLE II



*„Ich besaß als Jump-Man drei (Spieler)-Leben. Doch scheinbar war bei meinen Aktionen das Risiko zu groß!“*

Achten Sie darauf, daß es Ihnen nicht ebenso wie unserem Helden ergeht!

## Spielbeschreibung

10 Sekunden nach dem Starten des Programms wird das Titelbild aufgebaut. Nach dem Drücken einer Taste muß wieder ca. 20 Sekunden gewartet werden.

Es erscheinen nun 3 Reihen Luftballons, ein Männchen auf einem Podest und ein Männchen auf einer Wippe. Durch Drücken einer Taste springt das Männchen von dem Podest und fliegt in einer Parabel nach unten. Die Wippe ist nun mit den Tasten „L“ für Links und „+“ für Rechts so zu be-

wegen, daß das fliegende Männchen auf die freie Wippenseite trifft. Angehalten werden kann die Wippe durch Betätigen der Spacetaste.

Es gibt nun zwei Arten von Sprüngen:

a) einen Flach-weit Sprung. Die Wippe muß hier durch das Männchen möglichst nahe am Wippenmittelpunkt getroffen werden.

b) einen Hoch-kurz Sprung, wobei das analoge zu a) gilt. Die Wippe muß hier

## „Jump sorgt für Action, Spannung und Aufregung

möglichst fern vom Wippenmittelpunkt getroffen werden.

Durch eine geschickte Aneinanderreihung von Flach-weit- und Hoch-kurz Sprüngen kann somit jede Bildschirmposition vom Männchen erreicht werden. Die oberste Ballonreihe kann nur mit einem Hoch-kurz Sprung erreicht werden. Je nach Sprungart gibt es beim Auftreffen des Männchens auf der Wippe einen höheren bzw. einen niedrigeren Ton. Wird nun ein Ballon vom Männchen getroffen, so fliegt dieser daraufhin senkrecht herrunter.

Der Spieler erhält je nach Lage des Ballons von unten nach oben 50, 100, 150 Punkte und einen entsprechend hohen Ton.

Hat man alle Ballons abgeräumt, so bekommt man 10% der erreichten Punkte als Bonus.

Sind alle 3 Männchen aufgebraucht, so erscheint das Schlußbild, wobei je nach Punktzahl eine ansteigende bzw. abfallende Tonfolge erklingt. Durch Drücken der „y“-Taste wird das Spiel neu gestartet.

## Besonderheiten:

In Zeile 1720 - 1770 wird zwischen HPAG1 und HPAG2 alternierend hin- und hergeschaltet, um die Männchen zum Laufen zu bringen.

Die Flugparabel entspricht einer Erdbeschleunigung von ca. 1,3 g.

Um eine lange Rechenzeit im Programm zu vermeiden, wurden die Flugpositionen vor dem eigentlichen Programm berechnet. Sie werden dann nur noch aus einem Feld heraus gelesen, wenn sie benötigt werden.

```

10 REM *****
20 REM *      JUMP-MAN      *
30 REM *    FOR APPLE II   *
40 REM *      BY           *
50 REM *    RUDI NEUMANN   *
60 REM *    6203 HOCHHEIM/M *
70 REM *    MARZELSTR 38   *
80 REM *****
90 HCOLOR= 3: SCALE= 1: ROT= 0
100 GOSUB 2350
110 GOSUB 1790

```

```

120 GOSUB 1360
130 GOTO 1240
140 XDRAW WS(F) AT WX,190:WX = 5
    O:OX = 50:F = 0: XDRAW WS(F)
    AT 50,190
150 POKE 768,150: POKE 769,100: CALL
    770
160 POKE 768,200: POKE 769,100: CALL
    770
170 POKE 768,250: POKE 769,100: CALL
    770: CALL 770

```



```

180 MA = MA - 1: IF MA = 0 THEN GOTO
    1870
190 X = PX(29,2) - 102: Y = PY(29,
    2)
200 A = 2: R = 1: Q = 1: AN = 0
210 XDRAW 4 AT X,Y
220 HPLLOT 0,82 TO 10,82 TO 10,85
    TO 0,85
230 POKE - 16368,0: GET A#
240 HCOLOR= 0: HPLLOT 0,82 TO 10,
    82 TO 10,85 TO 0,85: HCOLOR=
    7
250 Z = 28
260 DX = - 102
270 AX = X: AY = Y
280 REM
290 REM HAUPTPROGRAMM
300 REM
310 IF PEEK (49152) = 187 OR PEEK
    (49152) = 204 THEN GOSUB 49
    0
320 GOSUB 360
330 IF PEEK (49152) = 187 OR PEEK
    (49152) = 204 THEN GOSUB 49
    0
340 GOTO 310
350 REM
360 REM MAENNCHEN-BEWEGUNG
370 REM
380 Z = Z + R
390 IF Z = 41 OR Z = - 1 THEN XDRAW
    4 AT AX,AY: POP : GOTO 140
400 MX = PX(Z,A * Q) + DX: MY = PY
    (Z,A)
410 IF MX < 0 THEN MX = MX + 279
420 IF MX > 279 THEN MX = MX - 2
    79
430 XDRAW 4 AT AX,AY
440 XDRAW 4 AT MX,MY
450 AX = MX: AY = MY
460 IF PEEK (234) < > 62 THEN
    GOSUB 590
470 RETURN
480 REM
490 REM WIPPEN-BEWEGUNG
500 REM
510 WX = WX + 4 * ( PEEK (49152) =
    187) - 4 * ( PEEK (49152) =
    204)
520 IF WX < 0 THEN WX = 279
530 IF WX > 279 THEN WX = 0
540 XDRAW WS(F) AT DX,190
550 XDRAW WS(F) AT WX,190
560 DX = WX
570 RETURN
580 REM
590 REM MAENNCHEN-KOLLISION
600 REM

```

```

610 IF MY < 100 THEN GOTO 780
620 XDRAW WS(F) AT WX,190: F = NOT
    F
630 XDRAW WS(F) AT WX,190
640 IF AN = 0 THEN R = - R: AN =
    1: Z = 2
650 R = - R
660 D = WX - MX
670 IF D < - 12 OR D > 12 THEN
    700
680 IF (D < = - 6 AND R = 1) OR
    (D > 6 AND R = - 1) THEN
    A = 1: GOTO 710
690 IF (D < = 0 AND R = 1) OR (
    D > 0 AND R = - 1) THEN
    A = 2: GOTO 710
700 XDRAW 4 AT AX,AY: POP : POP
    : GOTO 140
710 DX = WX - R * 10 - PX(Z,A) *
    (R = - 1)
720 POKE 234,0
730 Q = 1
740 POKE 768,50 * A: POKE 769,25

750 CALL 770
760 RETURN
770 REM
780 REM BALLON LOESCHEN
790 REM
800 BY = 15
810 IF MY > 51 THEN BY = 59: GOTO
    830
820 IF MY > 29 THEN BY = 37: GOTO
    860
830 RX = INT ((MX - 14) / 18 + .
    5)
840 BX = RX * 18 + 14
850 GOTO 880
860 RX = INT ((MX - 4) / 18 + .5
    )
870 BX = RX * 18 + 4
880 XDRAW 3 AT BX,BY
890 Z = 40 - Z
900 Q = 0
910 DX = MX
920 BZ = BZ - 1: IF BZ = 0 THEN GOTO
    2190
930 SC = SC + (3 - (BY - 15) / 22
    ) * 50
940 POKE 768,2 * BY: POKE 769,25

950 CALL 770
960 RETURN
970 REM
980 REM POSITIONS-BERECHNUNG
990 REM
1000 FOR % = - 10 TO 10 STEP .5
1010 Y = INT (13 * X ^ 2 / (2 *

```





```
      2 ^ 2))
1020 PX(Z,1) = INT (X * 5 + 50)
1030 PY(Z,1) = Y + 24
1040 Z = Z + 1
1050 NEXT
1060 Z = 0
1070 FOR X = - 14.5 TO 14.5 STEP
      .723
1080 Y = INT (13 * X ^ 2 / (5 *
      2 ^ 2))
1090 PX(Z,2) = INT (X * 5 + 76)
1100 PY(Z,2) = Y + 50
1110 Z = Z + 1
1120 NEXT
1130 RETURN
1140 REM
1150 REM BALLONS-INITIALISIEREN
1160 REM
1170 FOR Y = 0 TO 2
1180 FOR X = 4 TO 256 STEP 18
1190 XDRAW 3 AT (Y < > 1) * 10 +
      X,15 + Y * 22
1200 NEXT X,Y
1210 XDRAW 3 AT 4,37
1220 RETURN
1230 REM
1240 REM INITIALISIERUNG
1250 REM
1260 DIM PX(40,2),PY(40,2)
1270 GOSUB 980
1280 WS(0) = 1:WS(1) = 2
1290 WX = 50:OX = 50
1300 MA = 4:SC = 0:BZ = 44:F = 0
1310 HGR2
1320 XDRAW 1 AT WX,190
1330 GOSUB 1170
1340 GOTO 180
1350 REM
1360 REM TITELBILD
1370 REM
1380 HGR : POKE - 16302,0: HGR2

1390 FOR I = 0 TO 3
1400 READ LE
1410 FOR L = 1 TO LE
1420 READ X,Y
1430 Y = 50 + Y * 12
1440 X = 30 + X * 9 + I * 54
1450 Y1 = 180
1460 XDRAW 3 AT X,Y1
1470 FOR Y0 = 180 TO Y STEP - 4

1480 XDRAW 3 AT X,Y1
1490 XDRAW 3 AT X,Y0
1500 Y1 = Y0
1510 NEXT
1520 POKE 230,32: XDRAW 3 AT X,Y
      1: POKE 230,64
1530 NEXT
```

```
1540 NEXT
1550 FOR X = 5 TO 278 STEP 26
1560 XDRAW 4 AT X,177
1570 XDRAW 4 AT X,8
1580 NEXT
1590 FOR Y = 36 TO 163 STEP 28
1600 XDRAW 4 AT 5,Y
1610 XDRAW 4 AT 265,Y
1620 NEXT
1630 POKE 230,32
1640 FOR X = 18 TO 252 STEP 26
1650 XDRAW 4 AT X,8
1660 XDRAW 4 AT X,177
1670 NEXT
1680 FOR Y = 22 TO 163 STEP 28
1690 XDRAW 4 AT 5,Y
1700 XDRAW 4 AT 265,Y
1710 NEXT
1720 POKE - 16300,0
1730 POKE 768,150: POKE 769,50: CALL
      770
1740 POKE - 16299,0
1750 POKE 768,100: POKE 769,50: CALL
      770
1760 IF PEEK ( - 16384) > 128 THEN
      GET AQ$: RETURN
1770 GOTO 1720
1780 REM
1790 REM MUSIK
1800 REM
1810 FOR I = 770 TO 790
1820 READ DA
1830 POKE I,DA
1840 NEXT
1850 RETURN
1860 REM
1870 REM SPIELEND
1880 REM
1890 TEXT : HOME
1900 S1$ = "#####
      #####"
1910 S2$ = "#
      #"

1920 PRINT S1$
1930 FOR I = 1 TO 21
1940 PRINT S2$
1950 NEXT
1960 PRINT S1$
1970 VTAB 3: HTAB 15: INVERSE : PRINT
      "JUMP - MAN": NORMAL
1980 VTAB 8: HTAB 5: PRINT "HIGH
      - SCORE"
1990 VTAB 8: HTAB 25: PRINT "YOU
      R SCORE"
2000 LH = LEN ( STR$ (HI))
2010 LY = LEN ( STR$ (SC))
2020 VTAB 10: HTAB 11 - LH / 2: PRINT
      HI
2030 VTAB 10: HTAB 30 - LY / 2: PRINT
```



```
SC
2040 IF SC > HI THEN HI = SC: VTAB
15: HTAB 3: PRINT "GOOD BOY
! YOU'VE DONE HIGH - SCORE "

2050 VTAB 20: HTAB 10: PRINT "AN
OTHER GAME CHAMP ?";
2060 IO = 100: I1 = 200: ST = 1
2070 IF SC = > HI THEN IO = 200
: I1 = 100: ST = - 1
2080 FOR I = IO TO I1 STEP ST *
20
2090 POKE 768, I: POKE 769, 50: CALL
770
2100 POKE 768, I + 10: POKE 769, 5
0: CALL 770
2110 NEXT
2120 POKE 768, 70: POKE 769, 200: CALL
770
2130 POKE - 16368, 0: GET AN$
2140 IF AN$ = "Y" OR AN$ = "J" THEN
XDRAW WS(F) AT WX, 190: GOTO
1280
2150 HOME
2160 PRINT "GOOD BYE BOY"
2170 END
2180 REM
2190 REM ALLES ABGERAEUMT
2200 REM
2210 FOR H = 1 TO 125 STEP 10
2220 POKE 768, H: POKE 769, 50
2230 CALL 770
2240 POKE 768, 250 - H: POKE 769,
50
2250 CALL 770
2260 NEXT
2270 CALL 770: CALL 770
2280 SC = SC + INT (SC / 10)
2290 XDRAW 4 AT MX, MY
2300 GOSUB 1170
2310 BZ = 44
2320 POP : POP
2330 GOTO 190
2340 REM
2350 REM SHAPE INITIALISIEREN
2360 REM
2370 FOR I = 24576 TO 24828
2380 READ DA
2390 POKE I, DA
2400 NEXT
2410 POKE 232, 0
```

```
2420 POKE 233, 96
2430 RETURN
2440 REM
2450 REM DATA FUER SHAPE, TITELB
ILD UND MUSIK
2460 REM
2470 DATA 4, 0, 10, 0, 66, 0, 122, 0, 20
5, 0
2480 DATA 73, 63, 63, 63, 12, 45, 45, 2
8, 63, 12, 4, 24, 8, 72, 73, 73, 6
3, 23, 63, 23, 63, 23, 63, 63, 23, 63
, 23, 63, 23, 63, 7, 40, 229, 39, 37,
39, 45, 252, 223, 103, 109, 45, 60,
63, 103, 45, 229, 63, 40, 45, 60, 63
, 12, 45, 0
2490 DATA 73, 63, 63, 63, 12, 45, 45, 2
8, 63, 12, 4, 24, 8, 216, 219, 219, 2
19, 45, 21, 45, 21, 45, 21, 45, 45, 2
1, 45, 21, 45, 21, 45, 5, 56, 103, 37
, 39, 37, 63, 108, 77, 229, 255, 63,
44, 45, 229, 63, 103, 45, 56, 63, 44
, 45, 28, 63, 0
2500 DATA 192, 216, 1, 1, 5, 5, 5, 1, 1,
2, 3, 3, 7, 3, 7, 7, 7, 2, 5, 5, 5, 5,
1, 5, 2, 3, 7, 3, 7, 7, 7, 6, 5, 5, 5,
5, 5, 1, 5, 2, 3, 3, 7, 3, 7, 7, 2, 1,
1, 5, 5, 5, 1, 1, 2, 3, 3, 3, 7, 3, 3,
2, 1, 1, 1, 5, 1, 1, 1, 2, 3, 3, 3, 7, 3,
3, 3, 2, 0
2510 DATA 24, 8, 24, 8, 24, 8, 24, 8, 17
3, 63, 55, 45, 245, 255, 42, 45, 45,
21, 63, 63, 63, 191, 109, 45, 109, 1
73, 255, 251, 251, 59, 78, 9, 45, 17
3, 255, 59, 23, 173, 191, 45, 77, 9,
63, 44, 5, 56, 7, 0
2520 DATA 173, 48, 192, 136, 208, 5, 2
06, 1, 3, 240, 9, 202, 208, 245, 174
, 0, 3, 76, 2, 3, 96
2530 DATA 13, 1, 1, 2, 1, 3, 1, 4, 1, 5, 1
, 5, 2, 5, 3, 5, 4, 5, 5, 5, 6, 4, 7, 3, 7
, 2, 7
2540 DATA 17, 1, 1, 1, 2, 1, 3, 1, 4, 1, 5
, 1, 6, 1, 7, 2, 7, 3, 7, 4, 7, 5, 7, 5, 6
, 5, 5, 5, 4, 5, 3, 5, 2, 5, 1
2550 DATA 17, 1, 7, 1, 6, 1, 5, 1, 4, 1, 3
, 1, 2, 1, 1, 2, 2, 3, 3, 4, 2, 5, 1, 5, 2
, 5, 3, 5, 4, 5, 5, 5, 6, 5, 7
2560 DATA 15, 1, 7, 1, 6, 1, 5, 1, 4, 1, 3
, 1, 2, 1, 1, 2, 1, 3, 1, 4, 1, 5, 2, 5, 3
, 4, 4, 3, 4, 2, 4
```

**An alle Programmautoren! -- ● -- An alle Programmautoren!**

*Wir, der Tronic Verlag suchen weiterhin Programmbeiträge, Tips u. Tricks etc. für die Systeme C-64/VC-20/C-16/CPC-464/Apple/II-99 und ZX-Spectrum. Es winkt ein stattliches Honorar: pro veröffentlichte Seite 120,- DM!*

# Ein außerordentlich ideenreiches Spiel für den TI 99/4A mit Extended Basic und Joysticksteuerung

## SCOOTIE

### Spielaufgabe ★ Spielaufgabe ★ Spielaufgabe ★ Spielaufgabe

Es geht um Scootie, den glücklichen Hausbesitzer. Er hat beschlossen, sein Dach zu isolieren. Zu diesem Zweck hat er Isolationsmaterial im Keller gestapelt und will es nun nach oben tragen und am Dach anbringen (mit Feuerknopf). Seine Kinder rollen jedoch ständig irgendwelche Gegenstände auf den Flur, über die

Scootie hinwegspringen muß. Außerdem leidet Scootie an einer Magen-Darm-Grippe, und muß ständig (jedes Mal wenn im Programm der Schriftzug „Toilette“ auftaucht) das „Örtchen“ aufsuchen. Zu allem Unglück ereignete sich im Keller auch noch ein Wasserrohrbruch, durch den das Isolationsmaterial zerstört würde.

Deshalb muß Scootie nun auch Sandsäcke aus dem ersten Stock in den Keller tragen, um das Material zu schützen. Stürzt unser Held beim Transport, so wird das Material beim Herunterfallen zerstört. Damit das Spiel weitergehen kann, muß Scootie erst die Überreste mit dem Eimer zur Mülltonne transportieren.

### Eingabetips ★ Eingabetips ★ Eingabetips ★ Eingabetips

Zuerst muß das Programm „Scootie-Grafik“ abgetippt werden. Dieses wird dann mit „RUN“ gestartet. Die vom Programm generierte Grafik-Datei muß nun auf Band aufgenommen werden. Anschließend kann man „Scootie-Grafik“ löschen.

Jetzt ist das Hauptprogramm einzugeben. Dabei sollte man sehr sorgfältig

vorgehen, da später nach jeder Fehlermeldung die Grafik zurückgesetzt wird. Dann kann das Hauptprogramm gestartet werden. Anschließend muß die Grafik-Datei geladen werden. Achtung! Nachdem auf die Anweisung „PRESS CASSETTE START, THAN PRESS ENTER“ hin ENTER gedrückt wurde, darf die Grafik-Datei erst dann geladen werden, wenn der Bildschirm um eine

Zeile nach oben gerückt ist! Jetzt kann das Spiel begonnen werden. Das Spiel ist zu Ende, wenn der Wasserspiegel den Sandsackhaufen übersteigt, Scootie mehr als fünfmal abgestürzt oder das Dach komplett isoliert ist. Die Alpha-Lock-Taste muß natürlich ausgerastet sein.

### Und nun zum Listing

```
100 CALL MAGNIFY(3):: RANDOMIZE
110 DIM A$(4),AM$(7),C(1),XS(7),YS(7),EX
(5),SA(1),S(2)
120 FOR I=0 TO 7 :: READ XS(I),YS(I):: N
EXT I
130 FOR I=0 TO 5 :: READ EX(I):: NEXT I
140 C(0)=57 :: C(1)=33 :: SA$="000000007
EFFFF7E"
200 OPEN #1:"CS1",INPUT,SEQUENTIAL,INTE
RNAL,FIXED 65
210 FOR I=0 TO 7 :: INPUT #1:AM$(I):: NE
XT I
220 FOR I=0 TO 4 :: INPUT #1:A$(I):: NEX
T I
230 FOR I=32 TO 56 STEP 4 :: INPUT #1:B$
:: CALL CHAR(I,B$):: NEXT I
240 FOR I=64 TO 116 STEP 4 :: INPUT #1:B
$ :: CALL CHAR(I,B$):: NEXT I
250 CLOSE #1
300 CALL COLOR(2,2,1):: DISPLAY AT(1,12)
ERASE ALL:"SCOOTIE":TAB(12);"-----"
310 FOR I=0 TO 4 :: CALL CHAR(120+4*I,A$
(I)):: DISPLAY AT(6+2*I,2):I+1;"FQR" ::
CALL SPRITE(#I+1,120+4*I,7,33+16*I,81)::
NEXT I
320 DISPLAY AT(16,3):"6 FQR LEICHT" :: D
ISPLAY AT(18,3):"7 FQR SCHWER" :: DISPLA
Y AT(21,5):"BITTE WJHLEN"
```

### T e i l E i n s

```
330 HI=MAX(INT(P/14.4),HI):: DISPLAY AT(
24,1):"BESTLEISTUNG,";HI;"/"
340 CALL KEY(0,RET,STA):: IF RET>55 OR R
ET<49 THEN 340 ELSE SW=RET-49 :: CALL DE
LSPRITE(ALL):: CALL CLEAR
400 DEF HIND(T)=-SW*(SW<5)+INT(RND*T/250
)*(SW=5)*(T<1250)+ABS(INT(RND*5)*((SW=6)
OR((T>1250)AND(SW>4))))
410 S(0),S(1)=3 :: S(2)=2
420 SA(0),SA(1)=1 :: LE=5 :: T,HW,P,PL,F
,SN,K,K1=0
430 CALL CHAR(120,"010303030103070B0B0B
0B63E3C20008040C0C080C0E4E8D0C0C0C06060C
0E")
440 CALL CHAR(124,"01020303010327170B030
3030606030780C0C0C080C0E0D0D0D0D0607C3C0
4")
450 CALL CHAR(39,"0",107,"0",108,A$(HIND
(T)),136,A$(HIND(T)),140,A$(HIND(T)),128
,AM$(4),132,AM$(5))
460 FOR I=1 TO 11 :: CALL COLOR(I,1,1)::
NEXT I
470 FOR I=0 TO 4 :: CALL HCHAR(13+I,25+I
,89):: CALL HCHAR(18+I,10-I,88):: NEXT I
480 FOR I=0 TO 5 STEP 5 :: CALL HCHAR(12
+I,1,112):: CALL HCHAR(12+I,2,113):: CAL
L HCHAR(11+I,1,116):: CALL HCHAR(11+I,2,
118):: NEXT I
```



```

490 CALL TUER(57,81,1):: CALL TUER(33,12
1,0):: CALL TUER(233,81,1)
500 RESTORE 2610 :: FOR I=1 TO 22 :: REA
D Y3,X3,A,B :: CALL HCHAR(Y3,X3,A,B):: N
EXT I
510 CALL VCHAR(8,10,92,5):: CALL VCHAR(8
,11,93,5)
520 FOR I=1 TO 24
530 Y=INT(RND*3)+3 :: X=INT(RND*28)+3 ::
CALL GCHAR(Y,X,A):: IF A=40 THEN 530
540 CALL HCHAR(Y,X,40):: NEXT I
550 DISPLAY AT(1,1):"PUNKTE: 0";TAB(17);
"hhhhh"
560 CALL COLOR(1,7,1,2,1,13,3,2,1,4,2,1,
5,2,1,6,2,1,7,2,1,8,2,1,9,9,2,10,6,1,11,
3,1)
570 CALL SPRITE(#15,120,5,41,128,#16,36,
1,25,132)
580 FOR I=1 TO 3 :: READ A,B :: CALL SPR
ITE(#1,A,B,81,57,0,-12):: FOR N=1 TO 400
:: NEXT N :: NEXT I
590 CALL TUER(57,81,0)
700 E=-2
710 CALL JOYST(1,X,Y):: CALL PATTERN(#15
,122-X/2):: CALL MOTION(#15,0,X*2,#16,0,
X*2)
720 T=T+.3 :: IF X=0 THEN CALL KEY(1,RET
,STA):: IF RET=18 THEN ON E+4+SN*2 GOTO
1100,800,1200
730 IF F+4*(E+2)=0 THEN CALL COINC(#15,-
(E+2)*120+41,XS((E+3)*2),5,A):: IF A AND
((Y=-4)+(E=-3)) THEN 750
740 GOTO 710
750 CALL MOTION(#15,0,0,#16,0,0):: IF E=
-3 THEN 2210
760 V=-8 :: CALL MOTION(#1,0,V,#2,0,V,#3
,0,V):: GOTO 1010
800 FOR I=-8 TO 8 :: CALL MOTION(#15,I,0
,#16,I,0):: CALL SOUND(-200,600-ABS(I)*1
0,0)
810 CALL KEY(1,RET,STA):: IF RET<>18 OR
F=0 THEN 870
820 T=T+.7 :: CALL MOTION(#15,0,0,#16,0,
0):: CALL POSITION(#16,Y1,X1):: IF X1>24
0 THEN 870
830 X3=X1/8+1 :: Y3=Y1/8+2.5
840 CALL GCHAR(Y3,X3,Z):: IF Z<>34 THEN
870
850 CALL HCHAR(Y3,X3,40):: P=P+20 :: DIS
PLAY AT(1,8)SIZE(5):P
860 F=F-1 :: PL=PL+1 :: CALL CHAR(37,RPT
$("00",8-F)&RPT$("FF",F))
870 NEXT I
880 T=T+4.2 :: CALL MOTION(#15,0,0,#16,0
,0)
890 CALL POSITION(#15,Y1,X1):: CALL LOCA
TE(#15,41,X1,#16,25,X1+4)
900 IF PL<60 THEN 710
910 CALL DELSPRITE(ALL):: CALL COLOR(2,2
,1):: DISPLAY AT(10,13)ERASE ALL:"BRAVO"
920 DISPLAY AT(13,1):"SIE HABEN ALLE PLA
TTEN ANGE-BRACHT." :: DISPLAY AT(15,1):U
SING "SIE BENNTIGTEN #### SEK.":T

```

```

930 FOR G=1 TO 10 :: FOR N=20 TO 1 STEP
-1 :: CALL SOUND(-10,2000+10*N*N,0):: NE
XT N :: NEXT G
940 GOTO 300
1000 CALL CHAR(128,AM$(4),132,AM$(5))
1010 CALL POSITION(#15,Y1,X1):: N=128
1020 CALL JOYST(1,X,Y):: IF Y=0 THEN 102
0

```



*„Herr Müller, falls Sie Ihre Eintippbemühungen derart ermüden, daß Ihre Arbeitsleistung darunter leidet, dann möchte ich Ihnen einen Tip geben. Machen Sie's so wie ich und bestellen Sie die tollen TRONIC-Programme einfach über den Software-Katalog!“*

```

1030 T=T+.2 :: Y1=Y1-Y
1040 N=260-N :: CALL SOUND(-30,1000-5*Y1
,0):: CALL PATTERN(#15,N):: CALL LOCATE(
#15,Y1,73,#16,Y1-16,77)
1050 ON INT((Y1-42)/39)+2 GOTO 700,1020,
1300
1100 CALL POSITION(#15,Y1,X1):: IF X1>10
4 OR X1<64 THEN 710
1110 CALL CHAR(37,"00000000FFFFFFFF")::
CALL SPRITE(#16,36,3,Y1-16,X1+4)
1120 FOR G=1 TO 6 :: CALL SOUND(1,523,0)
:: CALL SOUND(1,698,0):: NEXT G
1130 F=4 :: T=T+1.32 :: GOTO 710
1200 CALL POSITION(#16,Y2,X2):: IF X2<14
1 OR X2>155 THEN 710

```





```

1210 X3=1+(SA(0)<SA(1)):: SA(X3)=SA(X3)+
1
1220 CALL COLOR(#16,1):: CALL HCHAR(22-I
NT((SA(X3)-.5)/2),X3+19,34-(-1)^SA(X3))
1230 CALL SOUND(10,523,0):: FOR G=0 TO 3
0 STEP 2 :: CALL SOUND(-10,2794-G*50,G):
: NEXT G
1240 P=P+30 :: DISPLAY AT(1,8)SIZE(5):P
:: SN=0 :: T=T+1.9 :: GOTO 710
1300 FOR I=1 TO 7 :: CALL JOYST(1,X,Y)::
CALL PATTERN(#15,122-X/2):: CALL MOTION
(#15,0,X*2,#16,0,X*2)
1310 T=T+.44 :: CALL COINC(#15,YS(E+3),X
S(E+3),5,A):: IF A THEN 1380
1320 CALL KEY(1,RET,STA):: IF E<1 THEN C
ALL POSITION(#15,Y1,X1):: IF X1<49 AND X
+RET+SN=18 THEN 2100
1330 IF RET=18 THEN 1400
1340 CALL COINC(ALL,A):: CALL COINC(#15,
YS(E+3),1,4,B):: IF A OR B THEN 1500
1350 NEXT I
1360 IF INT(RND*10)=9 THEN 1900
1370 GOTO 1300
1380 CALL MOTION(#15,0,0,#16,0,0):: ON I
NT(LOG((ABS(E+.75)+.25)*2)+1.99)GOTO 100
0,1350,2200,1940
1400 CALL MOTION(#16,-18,X*1.25,#15,-18,
X*1.25):: CALL SOUND(20,20000,0):: CALL
COINC(ALL,A):: IF A THEN 1500
1410 FOR N=-14 TO 17.5 STEP 3.5 :: CALL
MOTION(#15,N,X*1.25,#16,N-SGN(N),X*1.25)
:: CALL COINC(ALL,A):: IF A THEN 1500
1420 NEXT N :: CALL SOUND(10,-5,0)
1430 CALL MOTION(#15,0,0,#16,0,0):: CALL
POSITION(#15,Y1,X1):: CALL LOCATE(#15,Y
S(E+3),X1,#16,YS(E+3)-16,X1+4)
1440 T=T+1.9 :: GOTO 1350
1500 CALL MOTION(#15,0,0,#16,0,0):: CALL
CHAR(128,AM$(INT(6.5-X/8))): CALL PATT
ERN(#15,128)
1510 CALL POSITION(#15,Y1,X1):: CALL LOC
ATE(#15,Y1-6,X1):: CALL MOTION(#15,4,X,#
16,4,1.5*X)
1520 IF SN+F=0 THEN 1550
1530 CALL DISTANCE(#15,#1,X1):: CALL DIS
TANCE(#15,#2,X2):: CALL DISTANCE(#15,#3,
X3)
1540 A=ABS((X1<X2 AND X1<X3)-1):: B=(X3<
X1 AND X3<X2)+3 :: CALL DELSPRITE(A,B)
1550 CALL POSITION(#15,Y1,X1):: IF Y1<YS
(E+3)-1 THEN CALL SOUND(10,880,0):: GOTO
1550 ELSE CALL MOTION(#15,0,0)
1560 IF F+SN=0 THEN 1610
1570 CALL POSITION(#16,Y3,X3):: IF Y3<YS
(E+3)-2 THEN CALL SOUND(10,880,0):: GOTO
1570
1580 IF F THEN CALL CHAR(37,"000000000011
E0FFF",39,"000000000080402F")ELSE CALL C
HAR(37,"000000000000C3E7F",39,"0000000000
00COE")
1590 CALL MOTION(#16,0,0)
1600 CALL LOCATE(#15,YS(E+3),X1,#16,YS(E

```

```

+3),X3)
1610 FOR I=0 TO 20 :: CALL SOUND(-10,523
-8*I,1,784+8*I,1):: NEXT I
1620 LE=LE-1 :: IF LE<0 THEN 2260 ELSE C
ALL HCHAR(1,19+LE,32)
1630 T=T+4 :: IF F+SN=0 THEN 1300
1640 IF K=0 THEN CALL DELHIND(1,6-A-B,C(
.5+SGN(1-ABS(E-1.5))/2))
1650 CALL PATTERN(#15,122-X/2)
1660 F,SN=0 :: K=SGN(E-.5)
1670 X1=246 :: X2,Y2=1 :: GOSUB 1800 ::
CALL SOUND(30,523,3)
1680 CALL POSITION(#15,Y1,X1):: CALL HCH
AR((Y1+15)/8,32,2):: CALL SPRITE(#14,40,
3,Y1-1,X1-8):: CALL SOUND(20,698,0)
1690 X1=X3 :: GOSUB 1800 :: CALL DELSPRI
TE(#16):: CALL SOUND(20,-5,0):: CALL CHA
R(37,RPT$("0",33))
1700 X1=6 :: GOSUB 1800 :: CALL SPRITE(#
17,116,3,YS(E+3),1):: CALL HCHAR((Y1+7)/
8,1,32,2)
1710 FOR I=-8 TO 8 :: CALL MOTION(#17,I,
0):: CALL SOUND(-10,210-ABS(I),0):: NEXT
I
1720 CALL DELSPRITE(#17):: CALL HCHAR((Y
1+7)/8,1,116):: CALL HCHAR((Y1+7)/8,2,11
7)
1730 X1=241 :: GOSUB 1800 :: CALL SOUND(
30,698,3)
1740 CALL DELSPRITE(#14):: CALL HCHAR((Y
1+15)/8,32,114):: CALL SOUND(20,523,0)
1750 T=T+3.7 :: IF K1 THEN 1300
1760 X1=XS(2.5+1.5*K):: X2=XS(6-K):: Y2=
YS(6-K):: GOSUB 1800
1770 E=2*K-(B+1)*(K+1)/2 :: IF B+K=-2 TH
EN 2100 ELSE 2210
1800 CALL JOYST(1,X,Y):: CALL PATTERN(#1
5,122-X/2):: CALL MOTION(#14,0,X*2,#15,0
,X*2)
1810 T=T+.34 :: CALL COINC(#15,YS(E+3),X
1,5,A):: CALL COINC(#15,Y2,X2,5-10*(E<1)
,B):: CALL KEY(1,RET,STA):: IF RET+A+B<>
17 THEN 1800
1820 CALL MOTION(#14,0,0,#15,0,0):: RETU
RN
1900 IF K1 OR E=3 THEN 1300
1910 CALL MOTION(#1,0,0,#2,0,0,#3,0,0,#1
5,0,0,#16,0,0):: DISPLAY AT(24,1):"TOILE
TTE"
1920 FOR G=1 TO 15 :: FOR N=1 TO 3 :: CA
LL SOUND(-10,100+10*G*G*N,G):: NEXT N ::
NEXT G
1930 K1=E :: E=2-SGN(K1-.5):: CALL MOTIO
N(#1,0,V,#2,0,V,#3,0,V):: GOTO 1300
1940 CALL COLOR(#15,1,#16,1):: CALL TUER
(233,81,0):: DISPLAY AT(24,1):""
1950 FOR G=20 TO 0 STEP -1 :: CALL SOUND
(20,523,G):: CALL SOUND(20,659,G):: CALL
SOUND(20,784,G):: NEXT G
1960 FOR G=0 TO 20 :: CALL SOUND(-10,-5,
G):: NEXT G
1970 CALL TUER(233,81,1):: CALL HCHAR(11
,31,102):: CALL HCHAR(12,31,103):: CALL

```



```

COLOR(#15,5,#16,3+4*SN)
1980 E=ABS(K1)*(K=0)-2*ABS(K):: K1=0 ::
T=T+10
1990 GOTO 1300
2100 CALL MOTION(#1,0,0,#2,0,0,#3,0,0)
2110 CALL POSITION(#15,Y1,X1):: IF X1<24
THEN 2150 :: X3=INT(X1/8)+2 :: IF S(X3-
5)=0 THEN 2150
2120 CALL CHAR(37,SA$):: CALL HCHAR(12+(
S(X3-5)>2),X3,33.5+1.5*(-1)^S(X3-5)):: S
(X3-5)=S(X3-5)-1
2130 CALL SPRITE(#16,36,7,Y1-16,X1+4,0,0
):: SN=1 :: T=T+3
2140 FOR G=1 TO 2 :: FOR I=0 TO 16 :: CA
LL SOUND(-10,523+I*15,0):: NEXT I :: NEX
T G
2150 E=-2 :: V=-7 :: CALL MOTION(#1,0,V,
#2,0,V,#3,0,V):: GOTO 1300
2200 IF E=-2 OR E=1 THEN CALL MOTION(#1,
0,0,#2,0,0,#3,0,0)
2210 CALL CHAR(128,AM$(.5+ABS(E+.5)),132
,AM$(-1.5+ABS(E+.5)))
2220 IF INT(LOG(T)^4/190)=HW THEN 2280 E
LSE HW=INT(LOG(T)^4/190)
2230 IF HW=INT(HW/8)*8 THEN CALL HCHAR(2
3-HW/8,21,106,12):: CALL CHAR(107,"0")::
CALL HCHAR(22-HW/8,21,107,12):: GOTO 22
80
2240 CALL CHAR(107,RPT$("00",8-HW+INT(HW
/8)*8)&RPT$("FF",HW-INT(HW/8)*8))
2250 IF SA(1)*4>=HW THEN 2280 ELSE DISPL
AY AT(24,1):"DAS WASSER LJUFT QBER"
2260 FOR N=1 TO 3 :: FOR G=-30 TO 30 ::
CALL SOUND(10,523,ABS(G),659,ABS(G),784,
ABS(G)):: NEXT G :: NEXT N
2270 CALL DELSPRITE(ALL):: GOTO 300
2280 RV=SGN(ABS(ABS(E)-2)-.5)
2290 Y1=YS(E+3)-2*RV :: X1=XS(E+3)
2300 U=Y1-RV*40 :: O=Y1+RV*4
2310 R=-SGN(ABS(E+.5)-2)
2320 E=EX(E+3):: T=T+1.5
2330 IF E=-3 OR(E=1 AND K=0) THEN 2400
2340 IF K=0 THEN CALL DELHIND(3,1,C(ABS(
(E-2)/3))): T=T+2.8
2350 T=T+5.6 :: X2=C(INT(.5+SGN(E)/2))::
Y2=YS(E+3):: CALL TUER((X2),(Y2),1):: G
OSUB 2450
2360 RESTORE 2640 :: FOR I=1 TO 3 :: REA
D A,B :: CALL CHAR(A,A$(HIND(T)))
2370 CALL SPRITE(#I,A,B,Y2,X2,0,24*SGN(A
BS(E-.5)-1)):: GOSUB 2450 :: NEXT I
2380 CALL TUER(X2,Y2,0):: GOSUB 2450 ::
IF K+E=0 THEN E=0
2390 K=0 :: IF K1 THEN E=3
2400 V=8*SGN(ABS(E-.5)-1):: CALL MOTION(
#1,0,V,#2,0,V,#3,0,V)
2410 GOSUB 2450 :: GOTO 2410
2450 CALL JOYST(1,X,Y):: Y1=Y1+X*R :: IF
Y1=0 THEN Y1=Y1-X*R :: GOTO 2480
2460 IF X=0 THEN 2480 ELSE CALL SOUND(10
,500,0)
2470 X1=X1+X :: CALL PATTERN(#15,130-X/2
):: CALL LOCATE(#15,Y1,X1,#16,Y1-16,X1+4

```

```

+2*R)
2480 IF Y1=U THEN 2490 :: T=T+.28 :: RET
URN
2490 CALL LOCATE(#15,YS(E+3),X1,#16,YS(E
+3)-16,X1+4)
2500 IF E=-3 THEN 710 ELSE 1300
2600 DATA 33,161,189,81,73,81,1,81,225,1
21,71,121,233,81,32,81,1,2,-2,0,-1,-3
2610 DATA 2,1,95,32,8,1,94,32,13,1,94,24
,13,29,94,4,18,1,94,6,18,11,94,22,23,1,9
4,32,3,1,40,96
2620 DATA 3,3,34,28,4,3,34,28,5,3,34,28,
22,19,35,2,22,10,40,1,22,12,40,1,22,14,4
0,1,22,21,107,12
2630 DATA 11,31,102,1,12,31,103,1,12,32,
114,1,17,32,114,1,11,5,35,2,12,5,33,3
2640 DATA 108,14,136,13,140,7
2700 SUB DELHIND(AS,AS1,XT)
2710 CALL POSITION(#15,Y,M):: CALL TUER(
(XT),Y,1):: CO=0
2720 FOR I=AS1 TO AS+AS1-1 :: CALL POSIT
ION(#I,Y,X)
2730 CALL MOTION(#I,0,20*SGN(SGN((M<XT)+
.5)*(ABS(X-(M+XT)/2)-(M-XT)/2+.5))): NE
XT I
2740 FOR I=AS1 TO AS+AS1-1 :: CALL COINC
(#I,Y,XT,13,A):: IF A THEN CALL DELSPRIT
E(#I):: CO=CO+1 :: IF CO=AS THEN 2760
2750 NEXT I :: GOTO 2740
2760 CALL TUER((XT),Y,0)
2770 SUBEND
2800 SUB TUER(X,Y,Z)
2810 X=X/8+1 :: Y=Y/8
2820 CALL VCHAR(Y,X,97+3*Z,3):: CALL VCH
AR(Y,X+1,96+3*Z,3):: CALL HCHAR(Y+1,X,96
+5*Z):: CALL HCHAR(Y+1,X+1,98+Z)
2830 SUBEND

```

### „SCOOTIE“: Teil Zwei

```

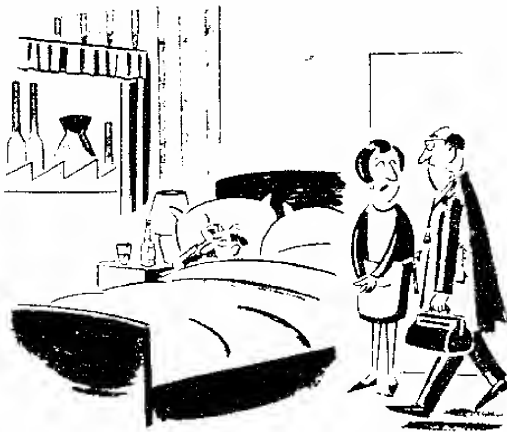
100 !*****
110 !* *
120 !* SCOOTIE-GRAFIK *
130 !* *
140 !*scootie&scootie-grafik*
150 !* von *
160 !* thorsten spott *
170 !* copyright(c) 15.IX.85*
180 !* *
190 !*****
200 OPEN #1:"CS1",OUTPUT,SEQUENTIAL,INTE
RNAL,FIXED 65
210 FOR I=1 TO 34 :: READ A$
220 PRINT #1:A$
230 NEXT I
240 CLOSE #1
250 !AM$(0):FIGUR OBERE TREPPE NACH OBEN
260 DATA 000000000000101020401030307000
060B0F0F060F0F8F4F4F4F4F03030307
270 !AM$(1):FIGUR UNTERE TREPPE NACH UNT
EN
280 DATA 060B0F0F060F1F1F2F4F0F0F0D0C0C1

```

# ... nebenbei bemerkt!

## Zur Genesung schenkt man oft Blumen ...

... und doch soll es einige Leute geben, die erst mit der **COMPUTRONIC** wieder so richtig auf die Beine kommen! Dazu meint Hausarzt Dr. Fiebertropf: „Ich verschreibe Ihrem Mann, Frau Bollerhey, noch schnell ein paar Aspirin, verordne ihm 24 Stunden Bettruhe, und kann dem Armen nur wärmstens ein Computronic-Abo empfehlen.“



Wußten Sie eigentlich, daß das Abo mit 6 Ausgaben pro Jahr nur 35 Mark - (im Ausland 45 Mark) - kostet?

Nun, Erna Bollerhey weiß es jetzt. Sie hoffentlich auch!!!

Die Abo-Karte finden Sie auf Seite 44!



TEXAS  
INSTRUMENTS



```
C0000000000008080404000E0E02
290 !AM$(2):FIGUR OBERE TREPPE NACH UNTE
N
300 DATA 0000000000000101020200070704000
060D0F0F060F0F8F8F4F2F0F0B0303038
310 !AM$(3):FIGUR UNTERE TREPPE NACH OBE
N
320 DATA 060D0F0F060F1F2F2F2F2F0F0C0C0C0
E0000000000008080402080C0C0E
330 !AM$(4):FIGUR AUF LEITER
340 DATA 090B0B0B09070303030303030303030
380C0C0C088C8E0D0C0E0F03
350 !AM$(5):FIGUR AUF LEITER
360 DATA 010303031113171B03070F0C0000000
090D0D0D090E0C0C0C0C0C0C0C0C0C0C
370 !AM$(6):FIGUR GESTUERZT NACH RECHTS
380 DATA 00000000000000000000030707CFF8B
00000000000000000000000000000000000000
390 !AM$(7):FIGUR GESTUERZT NACH LINKS
400 DATA 0000000000000000070FFBF770708103
00000000000000000000000000000000000000
410 !A$(0):BALL
420 DATA 000000000000000000000000000000000
00000000000000000000000000000000000000
430 !A$(1):SKATEBOARD
440 DATA 000000000000000000000000000000FF10313
100000000000000000000000000000000000000
450 !A$(2):HUND
460 DATA 00000000000000000008080F0F1F1F342
40000000000000000001836FEFEF0E0505
470 !A$(3):SERVIERWAGEN
480 DATA 0000000000000000201F08080F081C0
80000000000000000000000000000000000000
490 !A$(4):KINDERWAGEN
500 DATA 000000003C7C7C7C7F7F7F300D16381
```

```
00000000000000608F0F0F0F06080D0381
510 !CHAR 32-35:LEERZEICHEN,SANDSAECKE,D
ACH,SANDSACK
520 DATA 00000000000000007EFFFF7E7EFFFF7E
E007E7E7E7E7E7E00000000007EFFFF7E
530 !CHAR 36-39:GELOESCHT FUER #16
540 DATA 000000000000000000000000000000000
00000000000000000000000000000000000000
550 !CHAR 40-43:#EIMER
560 DATA 000000000000000000000000000000000
00000000000000000000000000000000000000
570 !CHAR 44-47:DOPPELPUNKT,MINUS,PUNKT,
BINDESTRICH
580 DATA 00181800001818000000007E7E00000
000000000000000018180000626408102646
590 !CHAR 48-51:0,1,2,3
600 DATA 007CFEC6C6C6FE7C001838581818183
C007CFE067CC0FEFE00FEFC081C06FEFC
610 !CHAR 52-55:4,5,6,7
620 DATA 000C1C3C6C6FE0C0C00FEFEC0FC06FE7
C007CFEC0FCC6FE7C00FEFC0C1818303
630 !CHAR 56-59:8,9
640 DATA 007CFEC67CC6FE7C007CFEC67E06FE7
C
650 !CHAR 64-67: ,A,B,C
660 DATA 000000000000000000000038386C7CEEEEC
600FCFEE6FCE6FEFC003C7EF8E0F07E3C
670 !CHAR 68-71:D,E,F,G
680 DATA 00F8FCFEC6FEFCF800F8F8E0FCE0FEF
E00FEFEE0FCE0E0E0003C7EF8E0FE7E3A
690 !CHAR 72-75:H,I,AE,K
700 DATA 00EEEEFEFEFEFEFEFE007C7C3838387C7
CC6BA386C7EEEEEC600EEECF8F8F8ECEC
710 !CHAR 76-79:L,OE,N,O
```

```

720 DATA 00E0E0E0E0FEFEFE82BA7CFEEFE7C3
8008ECEEFE6E200387CFEEFE7C38
730 !CHAR 80-83:P,UE,R,S
740 DATA 00FCFEE6FEFCE0E0EE00EEEEFEFE7
C00FCFEE6FEFCE0E007CFEF07C1EFE7C
750 !CHAR 84-87:T,U,V,W
760 DATA 00FEFEFE3838383800EEEEFEFE7
C00EEEE6C7C38381000D6D6D6FEFEE82
770 !CHAR 88-91:UNTERE TREPPE,OBERE TREP
PE
780 DATA 0F0F0F0F0F0F0F0F0F0F0F0F0F0F0
F.
790 !CHAR 92-95:LEITER,LEITER,FUSSBODEN,
DACHFIRST
800 DATA FFFFC0C0FFFFC0C0FFFF0303FFFF030
3FFFFF0000000000000000FFFFF
810 !CHAR 96-99:TUER GESCHLOSSEN,TUER OF
FEN
820 DATA FFFFFFFFFFFFFFFFFFFFFF7F7F7FFF

```

**T  
I  
9  
9  
/  
S  
C  
O  
O  
T  
I  
E**

```

FFFFFFFFFFFF1FEFF0
830 !CHAR 100-103:TUERELEMENT, ,TOILETTE
840 DATA 808080808080808080808080C0C0C
000000001010141310D03FFFE7C3C3E3F
850 !CHAR 104-107:KOPF, ,WASSER,WASSER
860 DATA 1834183C7EBE8D00000000000000000
0FFFFFFFFFFFFFFFFF0
870 !CHAR 108-111:LOESCHEN FUER #1(HINDE
RNIS)
880 DATA 000000000000000000000000000000
00000000000000000000
890 !CHAR 112-115:MUELLEIMER,EIMER
900 DATA 3F2A2A2515151D1FE0A0A0204040C0C
01C22417F3E3E3E3E0
910 !CHAR 116-119:#MUELLEIMERDECKEL
920 DATA 00000000002053F000000000000000
0000000000000000E00

```

**ENDE DES LISTINGS**

**Tips u. Tricks - Tips u. Tricks - Tips u. Tricks - Tips u. Tricks - Tips u. Tricks**

## Optimierung von Joystickabfrage in BASIC-Programmen

## Jetzt auch für den VC-20 und den C-16/116!

Grundsätzlich sind die Joystickabfragen bei den gängigen C (C-64/VC-20/C-16/C-128) untereinander nicht kompatibel. Das Anpassen und Umsetzen in andere BASIC-Dialekte ist unter diesen Umständen recht mühsam und man wird wohl in der Regel völlig neue Programme schreiben müssen.

Der C-16 besitzt wie auch der C-64 zwei Joystickports.

Während man beim C-64 über Umweg-Routinen (Speicherplätze : PEEK (56320) JOY1 & (56321) JOY2 zur eigentlichen Abfrage kommt, wartet der C-16 mit nur einem Befehl (PRINT JOY (1),(2) auf.

Zusätzlich kann man mit dem eingebauten Assembler -TED- unter Benutzung der Werte \$FF08 / PEEK (65288) kleine MA-Programme einbinden. (Die übrige „Tastatur“ bleibt davon unberührt.)

Eine schnelle Kombination (BASIC+MA) zeigt das vorliegende Programm. (TEST-PROGRAMM)

Ein wenig aus dem üblichen Rahmen fällt der VC-20 mit seiner JOYSTICKABFRAGE. Hier werden unterschiedliche System-Routinen (Kontroll-Ports) zur Positionsbestimmung (Richtung & Feuerknopf) herangezogen, die dann mit geeigneter „Software-Unterstützung“ abgerufen werden können.

Um diese Programme weitgehend (Überschreiben durch BASIC!) zu schützen, legt man sie zweckmäßig in „freie“ Speicherbereiche. Dazu bieten sich z. B. die Plätze im Puffer (VC-20 = \$033C - 03FB /

\*828 - 1019 oder C16 = \$0332 - 03F2 / \*818 - 1010) an. Ebenfalls für sogn. USER-SOFTWARE finden wir für den C-16 noch folgende Plätze: \$065E - 06EB / 1630 - 1771.

In der ZERO-PAGE sind (VC 20 = \$00FA - FE / \*251 - 254 & C 16 = \$00D8 - 00E8 / \*216 - 232) noch einige BYTES für den Anwender frei.

```

1  REM    JOY-TEST    C-16/116
2  GOSUB  100          : REM    BASIC-LADER
3  SCNCLR
4  LET B = 820
5  POKE B,JOY (1)
6  SYS  1630
7  ON PEEK ( 216) GOSUB  10, 20, 30, 40, 50,
                                60, 70,80, 90
8  POKE 216, Ø
9  GOTO  5
10 PRINT " 1  NORD  " : RETURN
20 PRINT " 2  NO    " : RETURN
30 PRINT " 3  OST   " : RETURN
40 PRINT " 4  SO    " : RETURN

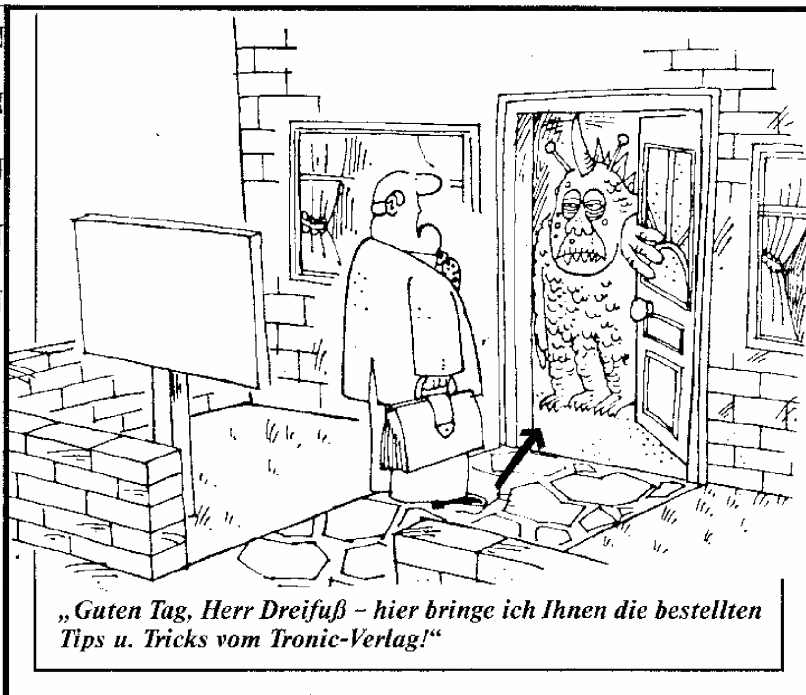
```



```

50 PRINT " 5   SUED   " : RETURN
60 PRINT " 6   SW     " : RETURN
70 PRINT " 7   WEST   " : RETURN
80 PRINT " 8   NW     " : RETURN
90 PRINT " 9   FIRE   " : RETURN
100 FOR I = 1630 TO 1648
110 READ N : POKE I, N : NEXT
120 RETURN
200 DATA 173, 52, 3, 201, 127,
      144, 7, 169
210 DATA 9, 133, 216, 76, 112,
      6, 41, 15, 133, 216, 96

```



```

1 REM JOY-TEST VC-20
2 REM BASIC -LADER
3 LET B = 829
4 FOR K = 0 TO 34
5 READ C : POKE B + K, C : NEXT
6 SYS 829 : REM AUFRUF
X REM
X Nach jedem Aufruf können über entspr. U-Prgr.
X PEEK(250) "FIRE"
X PEEK (253) " DIRECTION" (Richtungen)
X abgefragt werden
10 DATA 169,127,141, 34, 145, 173,32, 145,41, 128
20 DATA 133, 253, 169,255,141,34,145,173,31, 145
30 DATA 41,28,24,101,253,133,253, 173
40 DATA 31, 145, 41, 32, 133, 250, 96

*****

REM ! Der Recorder soll in "Rahestellung"
liegen.

```

**Wir bieten**  
*Action...*



*... lassen Sie zum  
Zug kommen ...*



*... verschenken  
so einiges ...*



*und berichten  
für Sie aus aller Welt*



**SA** aktueller  
software  
**markt**

das erste Wort in  
Sachen Software



# WAS GIBT'S NEUES AUF DEM C

## Textverarbeitung mit Home- und Personalcomputern

Wenn Computer-Käufer danach gefragt werden, wie sie ihre Rechner einsetzen, kommt prompt die klassische Antwort: neben Spielen und Programme abschreiben oder entwickeln „auch“ Texte schreiben. Das Verhältnis kehrt sich um im sogenannten professionellen Bereich, in dem Personal-Computer (PC's) eingesetzt werden. Dort wird zum großen Teil „nur“ geschrieben. Die Textverarbeitung gewinnt also immer mehr an Bedeutung.

Dem trägt das Buch von Alfred Görgens, „Textverarbeitung mit Home- und Personal-Computern: Systeme, Vergleiche, Anwendungen“, voll Rechnung. Der Autor, übrigens selbst ein erfahrener „Textverarbeiter“, greift wesentliche Fragen auf, die viele Anwender beschäftigen, und geht auf alle relevanten Punkte ein, wie den Kostenfaktor, die Leistungen einzelner Pro-

gramme sowie auf die Hardware, selbst mit ihrer Peripherie. Wer also nach dem geeigneten System, sucht oder erfahren möchte, ob Textverarbeitung überhaupt für ihn in Frage kommt, wer trotz kleinen Geldbeutels seine Texte lieber am Bildschirm gestalten möchte, wer einen bereits vorhandenen Personal-Computer für die Textverarbeitung einrichten lassen will, der findet in diesem Buch die praxisgerechten Tips und die aktuellen Vergleichsanalysen.

Die wichtigsten Rechner am Markt, vom Commodore 64 bis zum IBM AT, werden ebenso vorgestellt wie ihre Betriebssysteme und die dazu passende Textverarbeitungssoftware. Leicht verständliche Definitionen entwirren das ansonsten komplexe Fachvokabular. Ein praktischer Ratgeber, der schnell „Durchblick“ verschafft.



Dieser Buchtitel ist im Falken Verlag, Postfach 1120, 6272 Niedernhausen/Ts. erschienen, Preis: 29,80 DM

## „MICROSOFT WORD“



MICROSOFT WORD ist ein vielseitiges Textverarbeitungsprogramm von MICROSOFT, das professionellen Ansprüchen genügt. Seine Leistungsfähigkeit ist mit einer hohen Benutzerfreundlichkeit verbunden. Gleichgültig, um welche Schreibart es sich handelt; ob Briefe, Mitteilungen und Berichte, ob umfangreiche Dokumentationen oder wissenschaftliche Arbeiten, ob Rundschreiben oder Serienbriefe: MICROSOFT WORD macht die Texterstellung, das Redigieren und Archivieren müheloser und bietet vielfältige Gestaltungsmöglichkeiten.

Dieses offizielle Trainingsbuch ist als Lehrprogramm konzipiert, das der Benutzer in eigener Regie oder mit Unterstützung eines Schulungsleiters durcharbeiten kann.

Dem Buch ist eine Diskette als „Trainingssoftware“ beigegeben. Mit dieser Diskette

stehen dem Benutzer alle Funktionen des Programmes zur Verfügung, mit Ausnahme von Speichern und Druckern. Die Trainingssoftware enthält auch die Beispiele, die in diesem Lehrbuch durchgearbeitet werden. Es gibt keinen einfacheren und wirkungsvolleren Weg, um sich in MICROSOFT WORD einzuarbeiten, als dieses autorisierte, auch didaktisch und optisch besonders geschickt angelegte Buch mit der Trainingssoftware.

Textverarbeitung MICROSOFT WORD mit Trainingssoftware - „Softwareprübchen“, Carl-Hanser-Verlag, 119 Seiten; 1986. Gebunden 48,- DM.  
ISBN 3-446-14571-0

Der Autor des Werkes  
ist HELMUT SCHIRO

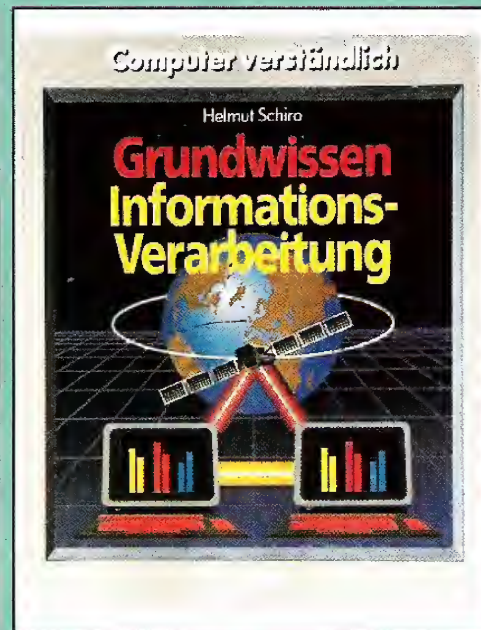


# COMPUTRONIC-BÜCHERMARKT ?

## Grundwissen Informations-Verarbeitung

Mit seinem Buch **Grundwissen Informations-Verarbeitung**, erschienen beim **Falcken-Verlag, Niedernhausen/Ts.**, räumt der erfahrene Autor **Helmut Schiro**, Berater beim größten Computerhersteller der Welt, mit einem leidigen Vorurteil gründlich auf: Computer und alles, was sich um sie und um die Datenverarbeitung dreht, müssen heute kein Buch mit sieben Siegeln mehr zu sein. Die erforderlichen Kenntnisse kann man sich leicht selbst aneignen! Manager, Entscheidungsträger in der Wirtschaft, Einkäufer von großen und mittleren Betrieben, sie alle können ohne weiteres in diese scheinbar komplizierte Materie einsteigen.

Das Buch von **Helmut Schiro** schafft ohne technischen Ballast die Voraussetzungen dafür. Dem Autor ist es gelungen, in einer klaren, für jederman verständlichen Sprache, auch komplizierteste Sachverhalte und Zusammenhänge einfach zu erläutern.



tern. Komplexe Fragen wie die Umwandlung von Zahlensystemen, die Struktur und der Aufbau von Datenverarbeitungsanlagen sowie die Verständigung zwischen Mensch und Maschine werden für den Leser transparenter, er versteht sie auf Anhieb.

Wer in seinem Betrieb die Einrichtung einer EDV-Anlage plant oder eine bereits vorhandene besser verstehen will, dem bietet dieses Buch das notwendige Grundwissen. Aber auch für Informatik-Studenten, Lehrer, Trainer, Berater und EDV-Neulinge wird dieser Buchtitel ein wertvolles Hilfsmittel im Umgang mit allen Aspekten der Informations-Verarbeitung sein. Die großzügige Bebilderung veranschaulicht Sachzusammenhänge, ein präzises Verweissystem führt den Leser durch das Buch, und ein Fachwörterverzeichnis im Anhang hilft, Computer-Chinesisch zu verstehen.

Preis: 58,-- DM

## Tabellenkalkulation „MICROSOFT MULTIPLAN“

Der Leser lernt das Tabellenkalkulationsprogramm „**MICROSOFT MULTIPLAN**“ kennen und bekommt aufgezeigt, wie man beim Aufbau von Modellen vorgeht, welche Gestaltungsmöglichkeiten für Modelle vorhanden sind und welche Nutzungsmöglichkeiten geboten werden. Dazu wird eine Fülle von Beispielen aus technischen und wissenschaftlichen bzw. kommerziellen Bereichen behandelt. Ziel dieser Lernbücher ist es, daß sich der Leser schrittweise zum **MULTIPLAN**-Anwender entwickelt, indem er die Beispiele durcharbeitet und am Rechner nachvollzieht.

Jedem Buch ist eine Diskette „Software-Probchen“ beigegeben, auf der eine „**MULTIPLAN**-Trainingssoftware“ sowie alle behandelten **MULTIPLAN**-Modelle gespeichert sind. Mit Hilfe dieser Diskette kann auf diese Modelle zurückgegriffen werden und es können damit die meisten der beschriebenen Tätigkeiten am Rechner ausgeführt werden.



Der Übende bekommt so einen realistischen Eindruck von **MICROSOFT-MULTIPLAN** und kann erkennen, inwieweit es für seine Aufgabenstellungen einsetzbar ist.

**MICROSOFT MULTIPLAN** mit Trainingssoftware - „Softwareprobchen“, **Carl-Hanser-Verlag**, 128 Seiten; 1986. Gebunden 48,-- DM.

ISBN 3-446-14573-7

*... bitte beachten:*

Alle hier vorgestellten Bücher können nicht beim **TRONIC-VERLAG** bestellt werden, sondern müssen über den **Buchhandel** oder direkt beim jeweiligen **Buch-Verlag** bezogen werden.



## **Disketteninhalt direkt vom Basic aus angezeigt**

Wenn man das Programm mit „RUN“ startet, schreibt es das File „DIR“ auf die Diskette in Station 1. Tippt man nun ENTER „D'DIR“ oder kurz E.“D:DIR“ ein, so wird das Directory der Diskette in Station 1 angezeigt, ohne daß das im Speicher stehende BASIC-Programm gelöscht oder verändert wird.

**Eine wertvoller Programmbeitrag für alle Atari-Besitzer**

```

1 REM *****
2 REM *
3 REM *   DIRECTORY VON BASIC   *
4 REM *           (C) 1986     *
5 REM *   VON OLIVER CYRANKA   *
6 REM *
7 REM *****
8 REM
9 DIM M$(200)
10 M$="CLR:DIM A$(20):? '5 NAME
11 M$?':? :CLOSE #1:OPEN #1,6,0,'D:*.':FO
12 R A=1 TO 70:INPUT #1,A$:? A$:"
13 M$(103)="IF A$(1,1)=' ' OR A$(1,1)='*
14 ' THEN NEXT A"
15 40, OPEN #1,8,0,"D:DIR":FOR N=1 TO LEN(M$
16 1:C=ASC(M$(N,N)):IF C=39 THEN C=34
17 50 PUT #1,C:NEXT N:PUT #1,155
18 60 CLOSE #1
    
```

*Die Tronic-Leserschaft sucht wieder den*

# **SOFTWARE-CHAMPION (1986)!**

**Wieder gilt es, einen „saftigen“ Geldpreis und wertvolle Sachpreise zu „erhaschen“. Wer sich auch in diesem Jahr am großen Rennen um die Meisterschaft beteiligen will, der sende sein Programm an die untenstehende Adresse (Abt. Programmentwicklung).**

Jedes eingesandte Programm nimmt an der Auswahl teil!!!  
Letzter Annahmetermin ist der 30. September 1986. Der Rechtsweg ist ausgeschlossen.

Nutzen Sie Ihre Chance, beteiligen Sie sich mit einem Software-Programm an unserem Wettbewerb!

### **Zur Auswahl:**

Wird ein Programm zur Darstellung im Heft herangezogen, zahlt der Verlag ein Honorar von DM 120,- pro abgedruckte volle Seite. Zum Jahresende wählt die Redaktion aus allen abgedruckten Programmen (HOMECOMPUTER, COMPUTRONIC, COMPUTE MIT) das Topprogramm des Jahres und ermittelt dadurch den „SOFTWARE-CHAMPION“ 1986. Unsere Leser und die Redaktion werden dann in der Ausgabe Dezember/Januar '87 das absolute TOPPROGRAMM des JAHRES wählen.

Der Autor des Programmes wird in unserer Ausgabe vorgestellt und mit einem Geldpreis fürstlich entlohnt.

**ACHTUNG:** Es werden nur Spielprogramme abgedruckt, bei deren Einsendung ein mit 1,40 DM frankierter Umschlag beigelegt wird!

**Einzusenden sind:**

++ Spielbeschreibung

++ Datenträger

++ Listing (nicht unbedingt erforderlich)

Einsenden an: TRONIC-VERLAG –

Postfach – 3440 Eschwege

**KENNWORT: SUPER-SOFTWARE-CHAMPION**

**Wir wünschen allen Freizeitautoren viel Spaß beim Mitmachen.**



## Achtung! – Rätselfreunde aufgepaßt – Achtung!

Dieses Mal haben wir uns etwas ganz Knif feliges ausgedacht: 10 Wörter sind in dem oben abgebildeten Buchstabenwirrwarr wiederzufinden. Um die Schwierigkeit der Rät selaufgabe noch zu erhöhen, werden die Wörter teilweise rückwärts oder diagonal eingetragen.

Markieren Sie bitte die gefundenen Begriffe und senden Sie die Lösung an den Tronic-Verlag, Am Stad 35 in 3440 Eschwege.

Auf die ersten 10 richtigen Einsendungen wartet wieder ein Datenträger aus unserem großen Softwareangebot (bitte Art des Da- tenträgers und Programmwunsch angeben!) Und nun wünscht Ihnen die Computronic-Redaktion viel Spaß beim „Tüfteln“!

**Folgende Begriffe sind zu suchen:**

PASCAL... BASIC... LOGO...  
FORTRAN . . . COBOL . . .  
FORTH... ASSEMBLER... PI-  
LOT... LISP... PROGRAM-  
MIERSPRACHEN.....

APTLIIJDSZBLNJYMW TZPNCSEIXJYSWECLURIWRZL  
VQCXVJNEHCARPSREIMMARGORPWXRZJNRUILIXFPW  
QUZKNPKLDUHUWIDLYWM SHUDHFACSBKHAHNNOQDRP  
VMLBCMSCMFWHCBITEBYMSNCDGLTOWNVOYVRUFMB  
SIJEFOBBNIEBFMYBATIBDXLRITOVUBSPFXBRYCOC  
ABFEHNLVRHYEGISOMQDRRNLIPLPNWHKITUIEBBYM  
ESHUGGANGFOYXFBCHRJVIMITIWKFTWSOACPEIPJ  
JNRBMBNOUNISRSLLDMCJBABPCM WUHPRFPEUZHHA  
LSYCLCCILSRKWDWEGJXKEPBLOMTNWAKIHFSETRQE  
UVLGAKNILFELPNWQIFUGLGFDC EORBRTRUAXSEZBI  
PVSOELHYIOFTUSRFVQMV RUOJVFPRELBMESSAVGUS  
AMEGHYQCXMFWDJWNUKENBSYFJSNFRKMVIBCWD PYG  
BBHYRCDSTAPASCALHEEBQKPTRRXKFHWYBGGVYLZM  
UTGKBLRPKOAMLQEEBYFKCGWAPUKXERMZCKNDCAOT  
KABWIAHTIIUAENOVPIFYGQNVWIMAAXIAVUYBTRO  
VABXBPHELENZHXOFABOBIKT BPMKRLPMEDDCVRWEP  
CCEPAITMOONEMA OVURRXFECLBKPUOSCPIEMUVJSI  
VGUSQOBXNURREWIOTJUJDGEIRVRVRUBHMB SAHMEG  
WUITKWZAZTBWYHAHLHREPBBSPYTRJBNRUZG IOSBO  
ECSRYSRQSQAJM PMJGEJGYERPRXLTTETWBERQJSHHT  
YLEAXTTAHRBRXRWEALUOMIBRVDEBASUPAJVLDIXC  
WKIRRYXIJHTPJVCUICWAKBBNJKDWABYLLEMTQVCM  
OPXOELPBBAMPGNHINWOOGOLIIFIWIRCOQRKDBOWB  
IUFVHEUHFSSJKNWIHUDAKANVHZE VVFGONS BHKIID

## Die Gewinner unseres Wettbewerbs „Überprüfen Sie Ihr Computerwissen:

Die Lösung des Wettbewerbs aus der Computronic-Ausgabe Nr. 3 lautete „AUTOREN“. Dieses Mal können sich folgende Gewinner auf Ihre Preise – Datenträger auf Wunsch aus unserem großen Softwareangebot – freuen: Der 1. Preis: (3 Kassetten) geht an Joachim Rösler, Mallinkrothstr. 5 in 4600 Dortmund 1. 2. Preis an W. Johannsen, Gustavstr. 12, Tübingen und Preis Nr. 3 an E. Wöller aus Mainz.

## VORSCHAU – VORSCHAU – VORSCHAU

**In der nächsten Computronic finden Sie:**

Zahlreiche und attraktive Programmbeiträge für die Computersysteme Commodore 64 / VC-20 / C-16 / Atari / Schneider / TI-99 / Apple und ZX-Spectrum.

Unter anderem erwarten Sie die Top-Programme

„Mister Postman“ (C-64) . . . , „Cosmic Terror“ (C-16) . . . ,  
„Smurf“ (Atari) . . . und „Star Patrol“ (CPC-464).

Weiterhin liefern wir Ihnen einen „Live“-Bericht vom 3. Wuppertaler Computer-Flohmarkt. – Außerdem setzen wir unseren beliebten Basic-Kurs fort und versorgen Sie mit zahlreichen weiteren interessanten Beiträgen!

## HINWEIS in eigener Sache!

### Liebe Computronic-Leser!

Viele von Ihnen werden sicherlich den Wettbewerb „Soundchampion des Monats“ vermissen. Dazu folgende Begründung: In den vergangenen Monaten wurden uns wiederholt raubkopierte Soundprogramme zugesandt und dies trotz mehrmaliger Hinweise, daß wir nur an ORIGINALPROGRAMMEN! interessiert sind. Aufgrund dieser Geschehnisse hat sich die Computronic-Redaktion schweren Herzens entschlossen, auf diesen Wettbewerb zu verzichten. – Teilen Sie uns doch Ihre Meinung dazu mit!

Die Redaktion



## C=64

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Mauern/Widerstand	8,--	15,--	C 41
Space-Comets/Erdsalte/Sprite-Data	15,--	23,50	C 51
Autostart/Bestellschein/Roadpainter	16,50	23,50	C 61
Hardcopy/Space-Fighter/ Data-Generator	15,50	19,50	C 71
Monster-Attack/Block-Painter/ Epson-Drucker	16,50	23,50	C 81
Projekt/Datenbank	16,--	23,50	C 91
Spiders/The Basic	16,50	23,50	C 101
High Noon/Skeet/Grafik/Designer	17,50	23,50	C 121
Painter/Star-Battle/Editor	17,50	23,50	C 22
Wüstenrallye/Jet-Pac/Black Moore Castle	17,50	23,50	C 32
Brieftaube/Cadelon	19,50	24,50	C 42
Ritter Erik/Grand Prix/Spritehilfe	19,50	24,50	C 52
Firebird/High Music	24,50	29,50	C 62
Moonsweeper/Scotti	24,50	29,50	C 72
Tron/Mercurious	24,50	29,50	C 13
Interceptor Base/Schotter	24,50	29,50	C 23
Jump Man/The Way/Space Taxi	24,50	29,50	C 33
Fight Night/Monsterjagd/Colossus Mühle	24,50	29,50	C 63

## VIC-20

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Bestellschein/Glücky	8,50	15,--	V 61
Multigraph/All Rammer	11,--	15,50	V 71
Zyklo/Meteorit	11,--	19,50	V 81
Garten/Schloß Gruselstein	14,--	19,50	V 91
Fressmann/Outlaw	14,--	19,50	V 101
Prost/Buffalo Bill	14,--	19,50	V 121
Joy Man/Powerpack	14,--	19,50	V 22
Der rasende Malocher	14,--	19,50	V 32
Matron/Obst	14,--	19,50	V 42
Race On/Cagy	14,--	19,50	V 52
Nürnberg - 3 D/Düsi	14,--	19,50	V 62
Breaker/Expulsion	14,--	19,50	V 72
Brückenbau/Jango	16,--	21,50	V 13
Miner, der Fassadenstreicher/Inkaschatz	16,--	21,50	V 23
Bergshooting/Booby in Action	16,--	21,50	V 33
Josef in den Katakomben	8,50	15,00	V 43

## C=16

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Horror Castle/Fantasy County	18,--	-	O 23
Goldrausch/Moon Fighter	18,--	23,--	O 33
Frogger/Humor	18,00	23,00	O 43

**Bitte beachten Sie:** Wenn Sie bei Ihrer Bestellung auf die sogenannte „Vorkasse“ zurückgreifen, so ersparen Sie sich viel Geld. Das gilt besonders für unsere ausländischen Freunde; da die Nebenkosten derart hoch sind, schicken wir keine Bestellung per Nachnahme ins Ausland. Schicken Sie uns deshalb einen Bar- oder Verrechnungsscheck, Postanweisung oder Bargeld. (Die Nebenkosten in der Bundesrepublik Deutschland - Nachnahme - betragen 5 DM)

## ZX81

Programm	Preis/ Kassette	Bestell-Nr.
Moon-Crash/ZX-Draw	10,--	Z 61
Tonprogramm/Aldebaran	10,--	Z 71
Reversi	10,--	Z 91
Panik Labyrinth	10,--	Z 101
Expedition	10,--	Z 121
Spinnen	14,50	Z 22
Spukhaus	14,50	Z 32
Frogger	14,50	Z 42
Olympiade	15,50	Z 52
Spider Mutants	15,50	Z 72
Der Elektroniker	10,--	Z 13

## TI-99/4A

**Achtung:** Die gesamten TI-Programme laufen nur mit Extended Basic

Programm	Preis/ Kassette	Bestell-Nr.
Drei-Kronen-Spiel/Zahlenputzen	8,50	T 41
Karl der Käfer/Alien-Landing	14,50	T 51
Jack the Digger I	14,50	T 61
Lift Bär/ASC II DEF Teil 1	14,--	T 71
Maya/ASC II DEF Teil 2	14,50	T 81
Transfer/Silverspar	14,50	T 91
Mother Duck/Screen Designer	14,50	T 101
Cave Man	8,--	T 121
Moon Race/Frogger/Slicks	19,50	T 22
Panzerschlacht	8,--	T 32
Maya II/Jagdszene Chicago	16,50	T 42
Raketen-Beschuß/Fire-Fox	16,50	T 52
Miner 99	14,50	T 62
Stardust	16,50	T 72
Jäger des verlorenen Schatzes	14,50	T 13
Thai-Boxing	12,50	T 23
Wanderer/Dessert Attack	16,50	T 33
Scoutie	14,50	T 43

## ATARI

### 800 XL (\*600)

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Mastermind*/Schlangenkrieg*	10,50	19,50	I 41
Tank-Battle/Oil Panic	12,50	-	I 51
Startup/Zeilen-Split/Tomstone-City	-	19,50	I 71
Painter/Hardcopy	14,--	19,50	I 81
The Big Quest/Fünf gewinnt	14,--	19,50	I 91
Split**	11,--	17,50	I 101
Ski/Mutation *	14,50	19,50	I 121
Super Miner/Diamonds	14,50	19,50	I 22
Donkey Kong/Kerzenheinz	16,50	19,50	I 32
Fighting/Escapes from Earth	16,50	19,50	I 42
Höhlen-Herbert/Ball Harbour	16,50	19,50	I 52
Aquonaut*	16,50	19,50	I 62
Jumping Ghost/Soundtrack	16,50	19,50	I 72
Groove	-	29,50	I 13 A
Treter/Cavehunt	11,--	16,--	I 13 B
Top Tennis/Panzerschlacht	16,50	-	I 23
Expedition/Cavehunt	-	21,50	I 33
Cavehunt	11,00	-	J 33 A
Boulder Dash	16,50	19,50	I 43





Programm	Preis/ Diskette	Bestell-Nr.
Wilder Westen/Karambolage/Maskengenerator	19,50	A 41
Music-Maker/Mission: Adler		
Disk-Katalog	19,50	A 51
Snake/Super Datei/Shape-tables	19,50	A 61
Library/Fight	19,50	A 71
Reversal/Disk-Menue-Generator	19,50	A 81
Diamonds/Hilfsprogramm	19,50	A 91
Tic-Tac-Toe/Jumper	19,50	A 101
Donovan/Basic-Konverter	19,50	A 121
Funktionstasten/Painter/Bowling	19,50	A 22
Thunder/Castle of Doom	19,50	A 32
Hubschrauber	19,50	A 42
Widerstandsdecodierung/Soft-Talker	16,50	A 52
Atlantic Gold/Vier gewinnt	16,50	A 72
Prodata/Music Editor	19,50	A 13
Pac Boy	16,50	A 23
Jump Man	14,50	A 33



Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
Super Miner	14,--	24,--	SR 41
CPC-Bert	14,--	24,--	SR 52
Conan's Castle	25,50	15,50	SR 62
Snider's Mace	16,50	26,50	SR 72
Schneider Panik/Killer Ship	18,50	28,50	SR 13
Midnight/Horror Caves	18,50	28,50	SR 23
Traumland/Triton	18,50	28,50	SR 33

**„Ich hab's Dir gleich  
gesagt, eine Bestellung  
aus dem Tronic-Software-  
Katalog hätte unser  
Freizeitproblem gelöst!“**

Wenn Sie derartige Streitgespräche vermeiden wollen, dann bestellen Sie unter folgender Anschrift:

**TRONIC-VERLAG**  
Am Stad 35  
3440 Eschwege



Programm	Preis/ Kassette	Bestell-Nr.
Inventur	12,--	S 41
Missile-Comment	8,50	S 51
Defender		
Lui der Wurm		
Alternativer Zeichensatz	13,50	S 61
Matheprogramm		
Bongo-Beeatcher	12,50	S 71
Solitaire		
Superstat		
Kleinstes gem Vielfache	14,50	S 81
Jump about	14,50	S 91
Pac-Man		
Oil Panic	14,50	S 101
Frogger	16,--	S 121
Jump	14,50	S 22
Jet Set Freddie	8,--	S 32
Andromeda	14,50	S 42
Totenkopf	14,50	S 52
Willibald	15,50	S 62
Squares	15,50	S 72
Karl der-Käfer	15,50	S 13
Otto Schweinsohr/Ausschnittkopierer	16,50	S 23
Car Crash/Cosmische Arche	16,50	S 33
Zone-M	15,50	S 43







# aktueller software markt

Nr. 5 Juni/Juli 1986

öS 50 · sfr. 6,- · DM 6,-

Anzeige

MASTERTRONIC

**Preisrätzel:**  
50 x 10 Kassetten  
sind zu gewinnen!

**Das aktuelle Software-Geschehen auf  
100 Seiten! Was man wissen sollte...**

★ Was hat GARFIELD, der berühmteste Kater  
der Welt, in der ASM zu suchen?  
Lesen Sie Seite 46

★ „LUCIFER'S REALM“ – das  
verbotene Spiel. ASM stellt  
es vor! Seite 88

★ „Volltreffer“ –  
Die High-Score-  
Liste Seite 23

**MIRRORSOFT**

**Quiz:**

115 Top-Games  
zu gewinnen!



... die  
interessante  
Seite mit  
Tips +  
Lösungen

**++ Action-Games ++ Anwender ++ Adventures ++ Sound- und Lernprogramme ++**

**HOTLINE** ... die  
Software-Hitparade

